**基于时空大数据的三化物联网云平台数据库设计说明书**

**所属项目：**

**文件类别：**

**版 本 号：**

**编 写 者：**

**审 核 者：**

**批 准 者：**

目 录

1引言 1

1.1编写目的 1

1.2背景 1

2外部设计 1

2.1标识符和状态 1

2.2使用它的程序 1

2.3命名约定 2

3结构设计 2

3.1概念结构设计 2

用户 2

租户组 3

客户组 4

设备 4

设备模型 5

设备组 5

报警规则 6

过滤器 6

插件 7

能力 7

属性数据 8

遥测数据 9

事件 9

数据模型 10

更新消息 10

仪表盘 11

部件实体 11

站点 12

近期数据 12

近期设备 13

设备接入ER图 13

3.2逻辑结构设计 14

user（用户） 15

user\_credentials（用户认证） 15

tenant（租户组） 15

tenants（3D平台租户） 16

customer（客户组） 16

device(设备) 16

devicesModel（设备模型） 17

device\_credentials（设备认证） 17

group（设备组） 17

device\_by\_group\_id（设备-设备组对应关系） 18

rule（报警规则） 18

Filter（过滤器） 18

rule2Filter\_Relation（规则与过滤器对应关系） 18

transform（插件） 18

rule2TransForm\_Relation（规则与过滤器对应关系） 19

ability（能力） 19

attributes\_kv\_cf（属性数据） 20

ts\_kv\_cf（遥测数据） 20

event（事件）： 21

data\_model（数据模型） 21

update\_message（更新消息） 21

dashboard（仪表盘） 21

widget\_entity（部件实体） 22

sites（站点） 22

recent\_data（近期数据） 22

recent\_device（近期设备） 23

4安全设计 23

4.1 Cassandra 23

4.1.1 TSL/SSL加密 23

4.1.2 Client authentication 24

4.1.3 Authorization 24

4.2 MySQL 24

4.2.1 安装最新的补丁 24

4.2.2 删除test数据库 25

4.2.3 控制可以访问所有数据库的账号 25

4.2.4 限制非管理员用户的权限 25

## 1引言

### 1.1编写目的

1. 本数据库设计说明书是关于基于时空大数据的三化物联网平台的数据库设计，主要包括数据逻辑结构设计以及运行环境、安全设计等。
2. 本数据库设计说明书读者：用户、系统设计人员、系统测试人员、系统维护人员。
3. 本数据库设计说明书是根据系统需求分析设计所编写的。
4. 本系统说明书为开发软件提供了一定基础。

### 1.2背景

根据国家“智慧城市”发展战略，华瀚科技早在2010年率先提出“智慧管网”理念，进行相应研究开发，并且把“智慧管网”定位为公司的核心发展战略；2013年，华瀚科技成立“锦瀚智慧管网技术有限公司”，专业从事“智慧管网”的研发和工程应用推广。

在国家十三五规划中，我国将投入15万亿元资金到国家基础设施建设中。在这个国家战略的大背景下，锦瀚智慧管网技术有限公司在市政业务上面临着极大的市场机遇。“智慧管网”系统平台将聚焦智慧水务、智慧管廊、智慧燃气、智慧安监、智慧园区、智慧建造、智慧建筑等核心业务的研究开发与应用推广，为建设智慧化城市综合管网提供完善的解决方案。为了实现这一目标，提取“智慧管网”应用系统的共性，能够将所有业务在一个统一的平台上运行。

## 2外部设计

### 2.1标识符和状态

数据库软件的名称：Cassandra，MySQL

数据库的名称为：BUPT\_IOT

### 2.2使用它的程序

1. 基础物接入
2. 物控制
3. 账户、认证及鉴权
4. 大数据服务
5. 3D WebGIS展示平台
6. 规则管理
7. 插件管理
8. 服务管理

### 2.3命名约定

所有的数据库表命名都是以具体表的英文词汇组成，这样能够统一数据库表的命名，也能够更好的规范数据库表命名。

## 3结构设计

三化物联网平台的数据库有两种，分别为Cassandra和MySQL。鉴于关系型数据库与非关系型数据库各自的特点，我们对不同的数据表单使用不同的数据库进行存储。

对彼此之间关系较弱且对写入速度要求较高的数据表单，包括事件、设备遥测数据与设备属性数据，会采用Cassandra数据库进行存储。为了方便，设备、设备证书与设备组我们也存储在Cassandra数据库中。

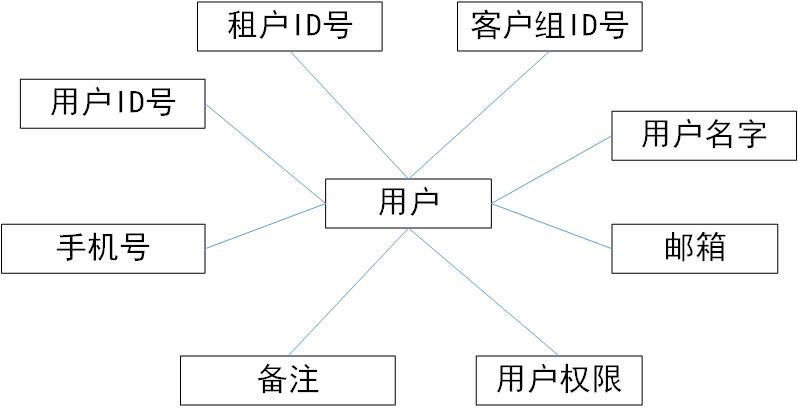
对于其他的数据表单（包括用户、租户组、客户组、服务、规则和插件等），对写入速度的要求较低，同时彼此之间拥有较高的关联性，故而采用MySQL关系型数据库进行存储。

### 3.1概念结构设计

分析三化物联网云平台系统的基本需求，利用概念结构设计的抽象机制，对所需分析结果中的信息进行分类、组织，得到系统的实体、实体属性、实体的键、实体之间的联系以及联系的类型，就可以设计出系统的概念模型。

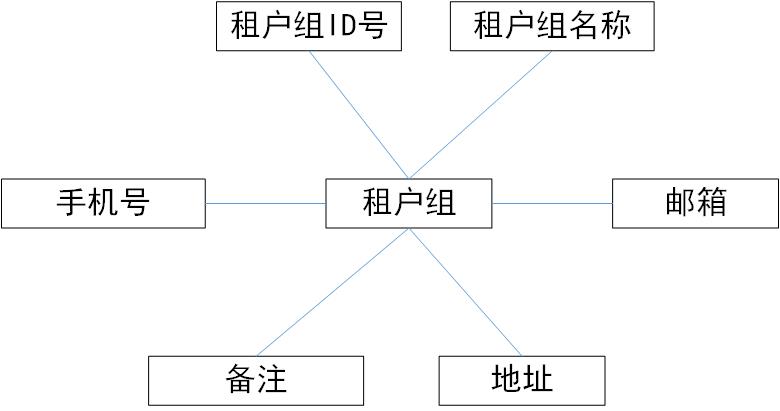
#### 用户

拥有用户ID号，租户ID号，客户组ID号，用户名字，邮箱，用户权限，手机号和备注。用户ID号用于辨识身份，租户ID号用于指明所属租户，客户组ID号用于指明所属客户组，用户权限用于标识该用于所拥有的权限级别，邮箱用于登录和接收警报信息，手机号用于接收短信。

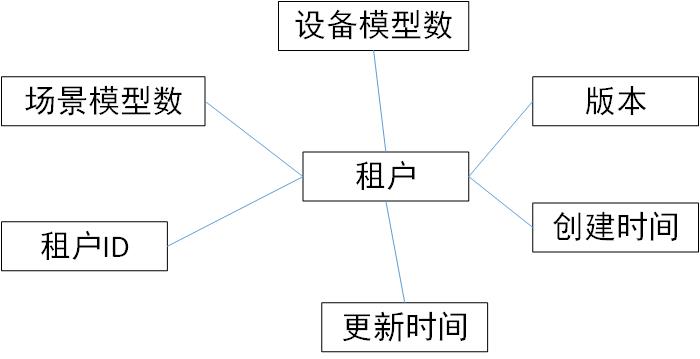


#### 租户组

拥有租户ID号，名称，邮箱，手机号和备注。ID号用于辨识身份，邮箱用于登录和接收警报信息，手机号用于接收短信。

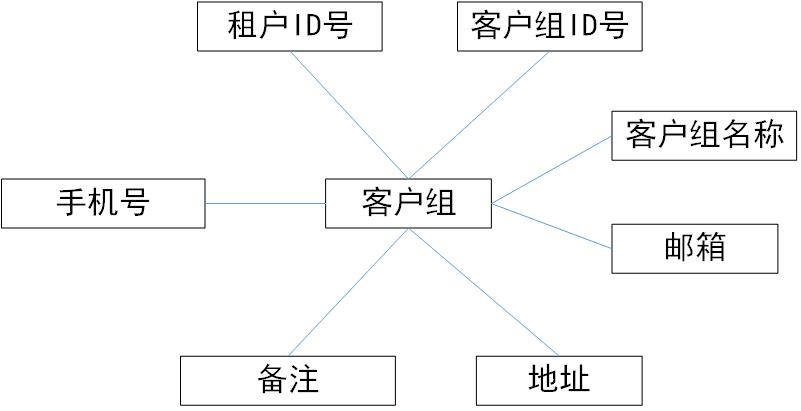


除了账户系统中的租户外，在3D平台也拥有一个租户表用于记录3D场景相关数据。拥有租户ID、场景模型数、设备模型数、版本、创建时间和更新时间等字段。租户ID用于标识某一个租户，场景模型数记录该租户下所有的场景模型的数量，设备模型数记录该租户下所有的设备模型数量。



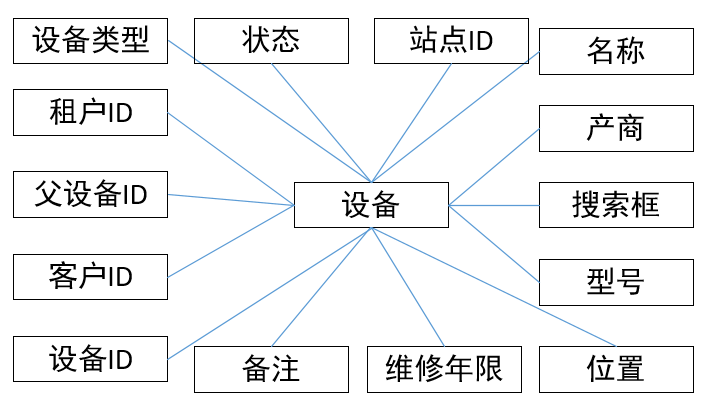
#### 客户组

拥有客户组ID号，租户ID号，邮箱，手机和备注。客户组ID号用于辨识身份，租户ID号用于指明所属租户，邮箱用于登录和接收警报信息，手机号用于接收短信



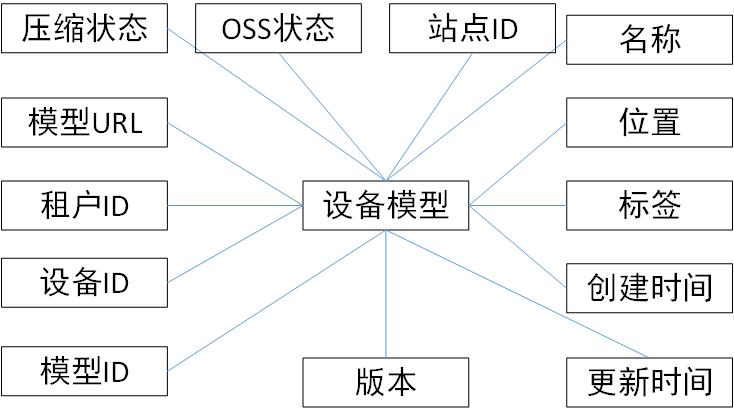
#### 设备

拥有设备ID号，设备名称，用户ID，租户ID，产商，位置，搜索框，状态，型号，设备类型和备注。设备ID号用于辨识设备，租户ID号用于指明所属租户，客户ID用于指明所属客户，产商、型号和设备类型该三元组用于确定所属服务组。设备位置用于指明目前设备所在的位置。父设备ID用于指明上一级设备。站点ID用于指明该设备被分配到哪一个地图场景中。维修年限指明设备可用期限。



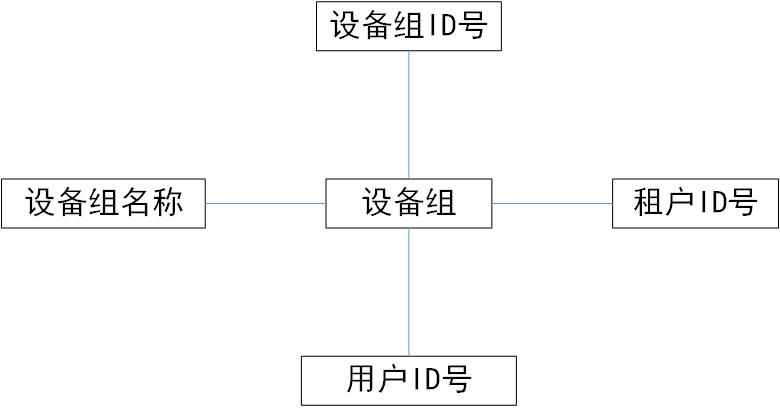
#### 设备模型

拥有设备ID、租户ID、、模型URL、压缩状态、OSS状态、站点ID、名称、位置、标签、创建时间和更新时间以及版本等字段。其中压缩状态和OSS状态目前作为保留字段以便日后升级。位置表明了该设备模型在场景中的位置。



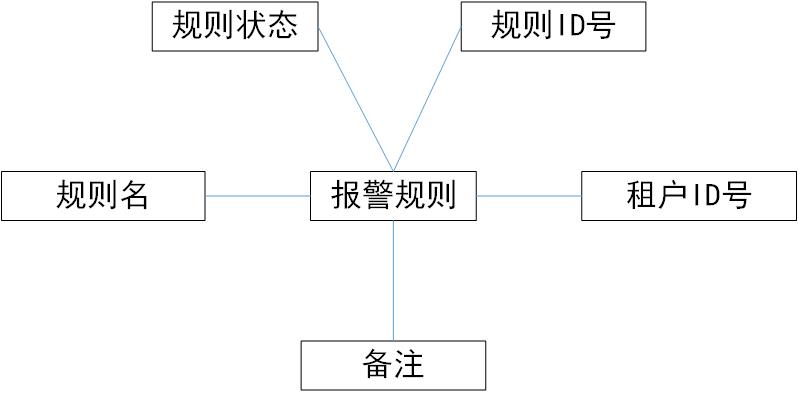
#### 设备组

拥有设备组ID号，租户ID号、用户ID和设备组名称。设备组ID号用于辨识设备组，租户ID号用于指明所属租户，用户ID用于指明所属用户。



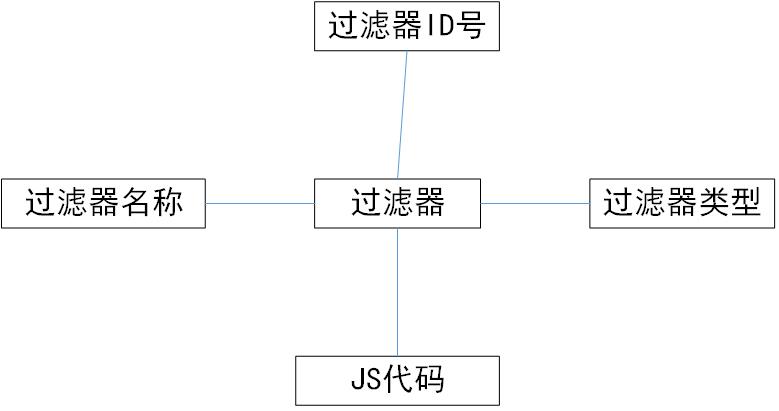
#### 报警规则

拥有规则ID、租户ID、规则名称、规则启用状态和备注。规则ID号用于辨识规则，租户ID号用于指明所属租户。规则启用状态有挂起和激活两种。



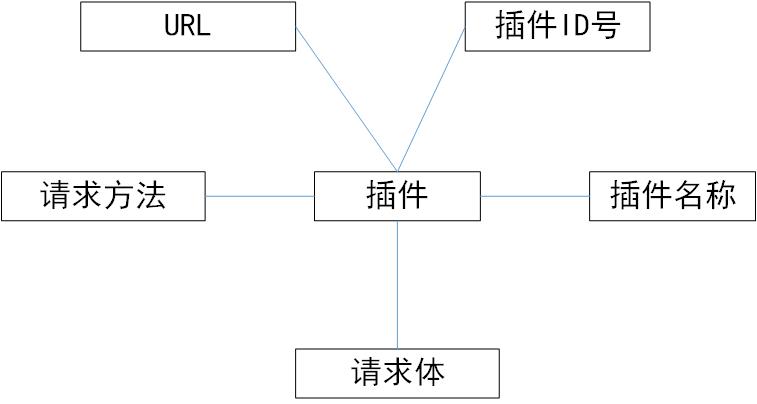
#### 过滤器

拥有过滤器ID，过滤器类型Js代码和过滤器名称。过滤器ID号用于辨识过滤器，过滤器类型用于指定过滤器所执行的不同过滤功能，js代码存储过滤器具体的过滤规则。



#### 插件

拥有插件ID号、插件名称、请求体、URL和请求方法。插件ID号用于辨识插件，请求体、请求方法和URL用于执行特定操作，完成插件功能。



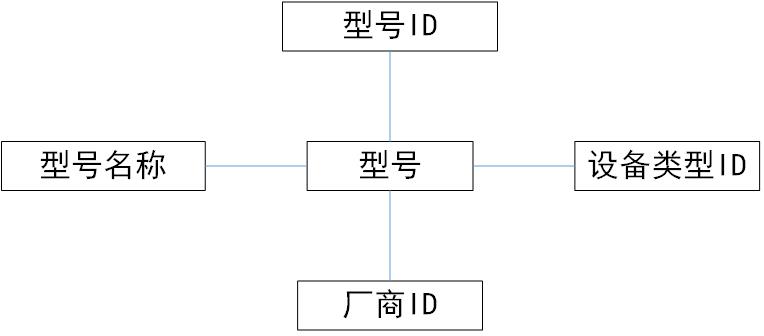
#### 能力

能力管理拥有厂商、设备类型、型号和能力四个表。

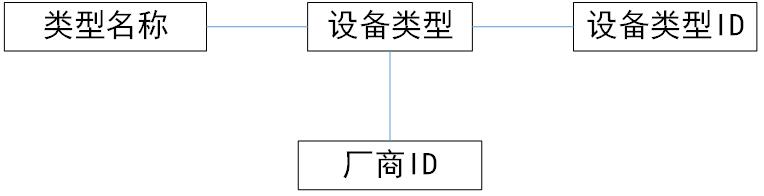
厂商：

厂商

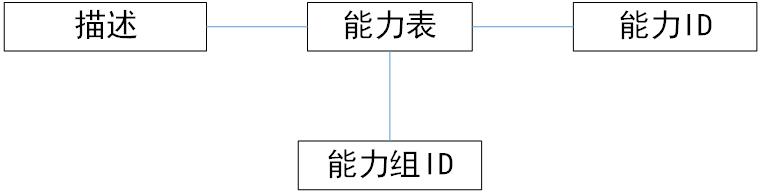
型号：



设备类型：

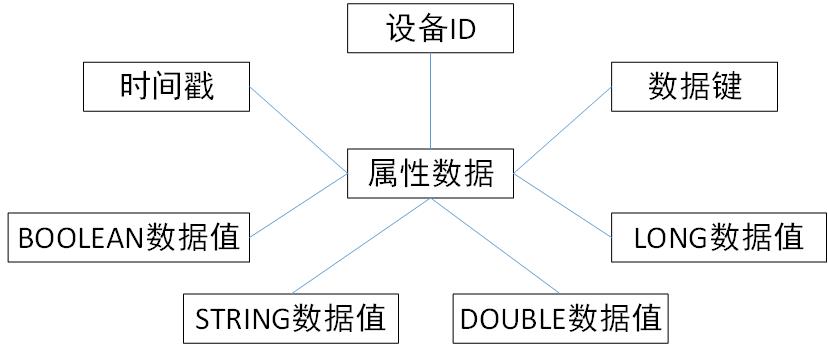


能力：



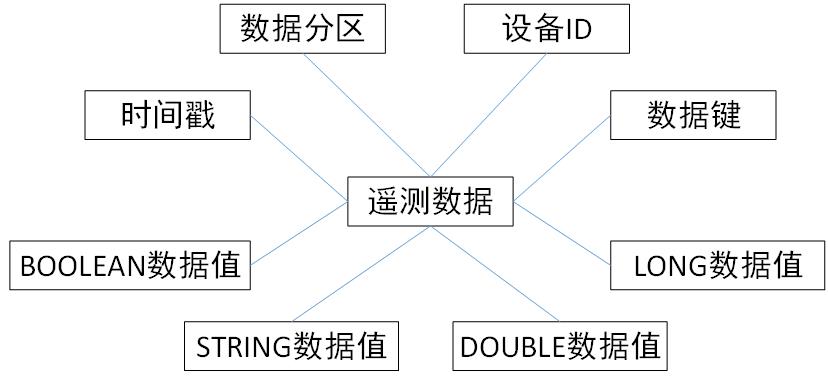
#### 属性数据

拥有设备ID号，属性键、属性值和时间戳。设备ID号表示该属性所属设备属性键表示属性键值对中的键，属性值为值，时间戳表示该属性最后更新的时间。



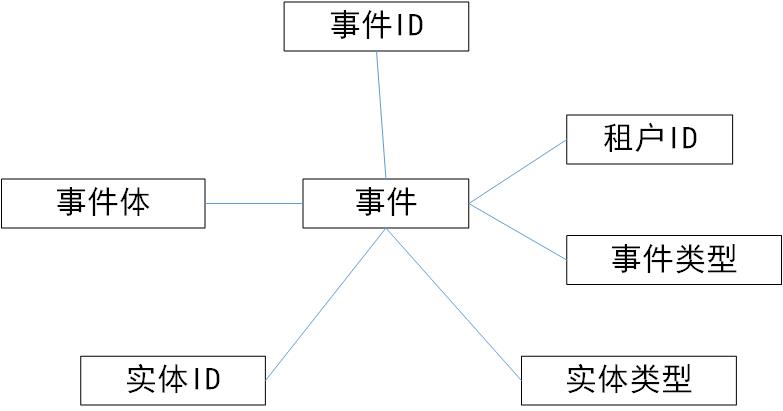
#### 遥测数据

拥有设备ID，数据键、数据值、数据分区和时间戳。设备ID号表示该数据所属设备，数据键表示遥测数据键值对中的键，数据值为值，时间戳表示该数据上传的时间。数据分区用于表明该数据所存储的分区，通常用一个LONG型时间戳来表示，可以设计一天或者一个月的数据量存储为一个分区。



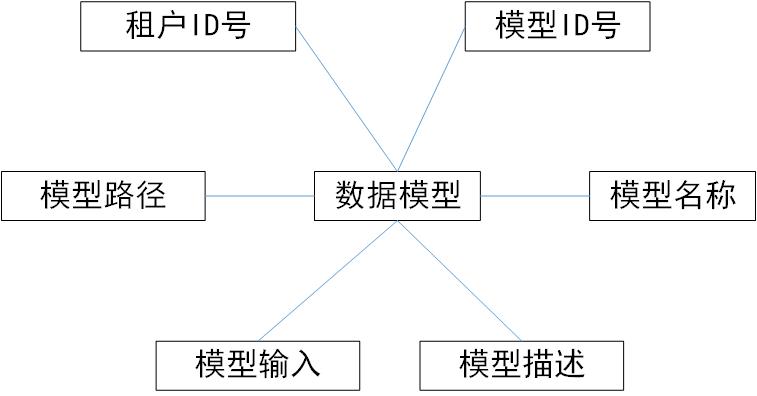
#### 事件

拥有事件ID，租户ID，事件类型，实体类型，实体ID和事件体等字段。事件ID用于区分不同事件，租户ID用于指明该事件所属的租户，事件类型可以自定义（例如报警事件，生命周期事件等），实体类型为该时间所属实体的类型（例如设备），实体ID用于指明具体实体，事件体保存事件的具体内容。



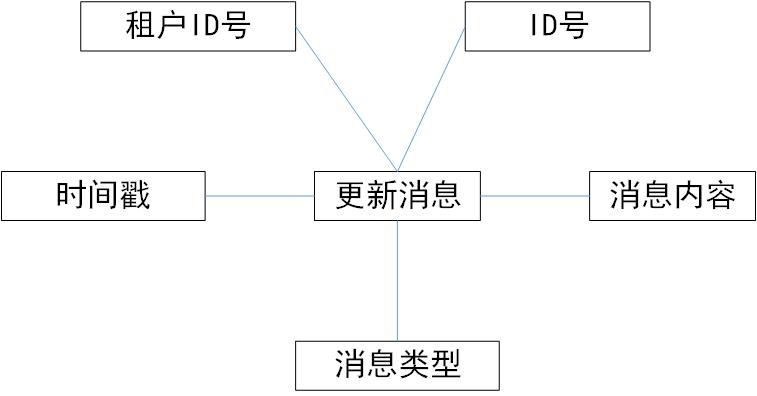
#### 数据模型

拥有模型ID，租户ID，名称、描述、模型输入和模型路径等字段。模型ID用于区分不同数据模型，租户ID用于指明该时间所属的租户，名称用于表示每一个模型，模型输入表示该模型要输入的数据，模型路径表示存储该模型的路径。



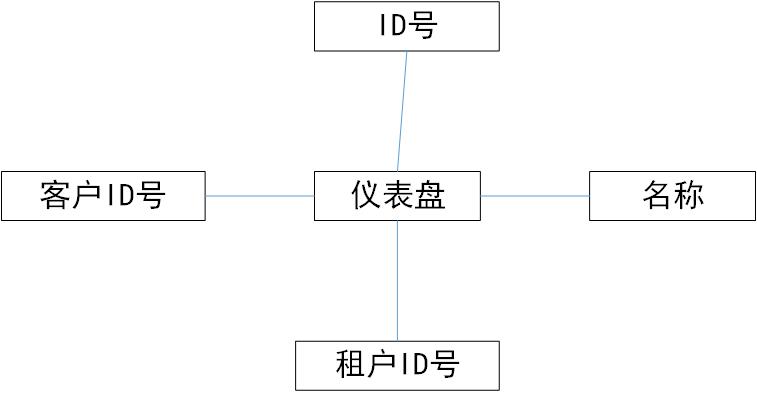
#### 更新消息

拥有ID，租户ID，消息内容、消息类型和时间戳等字段。ID用于区分不同的更新消息，租户ID用于指明该时间所属的租户，消息内容存储该消息的所要传递的信息，时间戳表示该消息生成的时间。



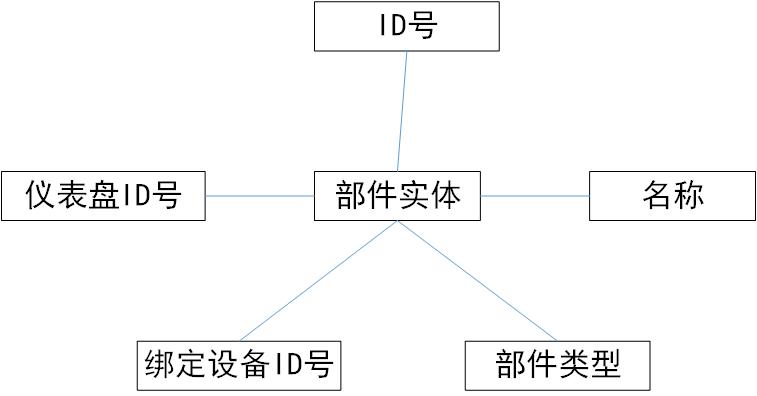
#### 仪表盘

拥有ID，租户ID，客户ID号和名称等字段。ID用于区分不同的仪表盘，租户ID用于指明该时间所属的租户，客户ID用于指明该时间所属的客户，名称用于标识每一个仪表盘。



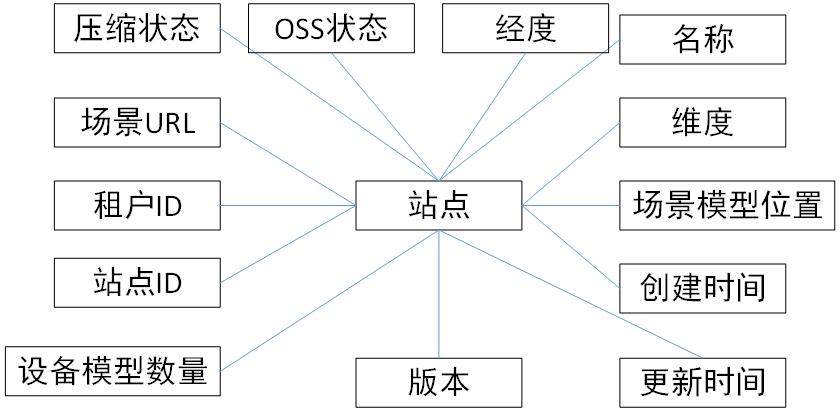
#### 部件实体

拥有ID，仪表盘ID，设备ID号、部件类型和名称等字段。ID用于区分不同的部件实体，仪表盘ID用于指明该部件所属的仪表盘，设备ID用于指明该部件所要展示的数据所属的设备，部件类型指明了该部件的类型（例如折线图等）。



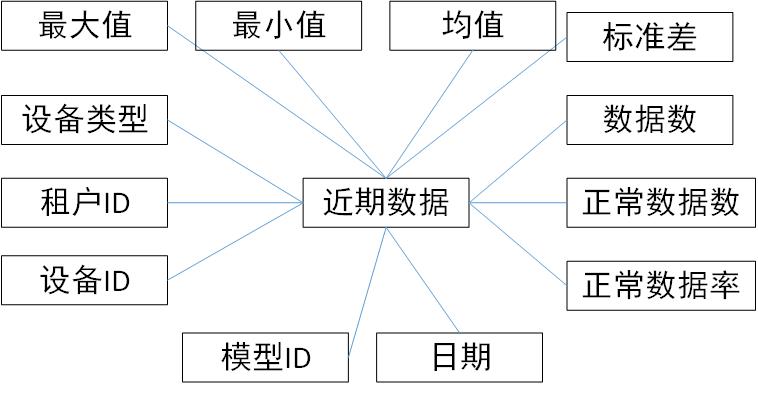
#### 站点

站点拥有站点ID、租户ID、场景URL、压缩状态、OSS状态、经纬度、名称、场景模型位置、创建时间、更新时间和版本。压缩状态和OSS状态作为保留字段以供日后升级使用。



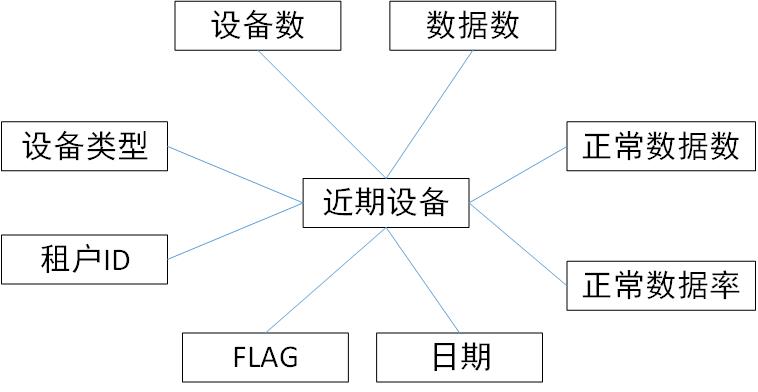
#### 近期数据

拥有设备ID、租户ID设备类型、最大值、最小值、均值、标准差、数据数、正常数据数、正常数据率、日期和版本。主要负责记录近期数据的各项数值指标。



#### 近期设备

拥有租户ID、设备类型、设备数、数据数、正常数据率、正常数据数、日期和FLAG。



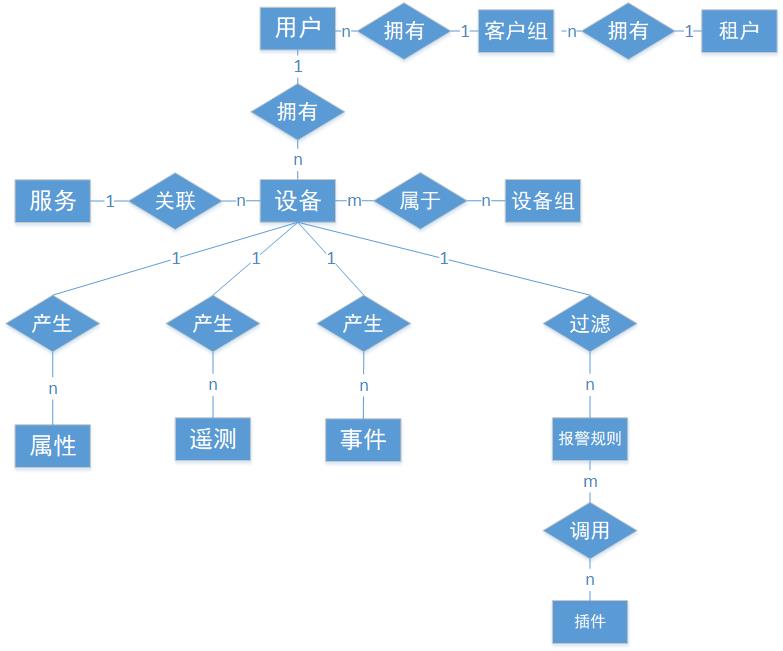
#### 设备接入ER图

一个用户拥有许多设备，二者为一对多关系。

设备通过厂商、类型和型号与唯一的服务进行配对，一个服务下能有许多设备。两者是一对多的关系。一个设备组拥有多个设备，一个设备可以分配到多个设备组。设备与设备组是多对多的关系。由于设备数量较多，设备与设备组的关系较为复杂，因此新引入一个关系表来表示设备与设备组两者之间多对多的关系。

一个设备能够产生多条遥测数据与不同的属性，而单一的一条属性或者遥测记录只能由一个设备产生。设备与属性、遥测是一对多的关系。设备产生的遥测数据过通过报警规则进行过滤，平台能为一个设备指定多条报警规则。两者是一对多的关系。设备与事件也是一对多的关系。当设备发生相应动作时，系统记录下该动作并存入数据库。

插件指定了在报警发生时，能够进行的操作。报警规则与插件通过报警规则的plugin\_token和插件的api\_token进行配对从而实现二者的关联。两者是多对多的关系。



### 3.2逻辑结构设计

关系模型的逻辑结构是一组关系模型的集合。E-R图则是由实体，实体的属性和实体间的联系三个要素组成。所以将E-R图转换为关系模型实际上就是要将实体，实体的属性和实体间的联系转换为关系模式。转换原则如下。

1. 实体类型的转换:一个实体型转换成一个关系模式。实体的属性就是关系的属性，实体的码就是关系的码。

2．联系类型的转换，根据不同的情况做不同的处理。

⑴ 一个1：1的联系可以转换为一个独立的关系模式，也可以与任意一端对应的关系模式合并。如果转换为一个独立的关系模式，则与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，每个实体的码均是该关系的候选码。如果与某一端实体对应的关系模式合并，则需要在该关系模式的属性中加入另一个关系模式的码和联系本身的属性。

⑵ 一个１：N的联系可以转换为一个独立的关系模式，也可以与N端对应的关系模式合并。如果转换为一个独立的关系模式，则与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，而关系的码为N端实体的码。

⑶ 一个M：N联系转换为一个关系模式。与该联系相连的各实体的码为各实体码的组合。

#### user（用户）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **id** | 唯一自增Id | int |
| **tenant\_id** | 租户Id | int |
| **customer\_id** | 客户idId | int |
| **email** | 邮箱 | varchar |
| **authority** | 权限 | varcahr |
| **name** | 名字 | varcahr |
| **additional\_info** | 备注 | varchar |

#### user\_credentials（用户认证）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **id** | 唯一自增Id | int |
| **user\_id** | 用户id | int |
| **password** | 用户密码 | varchar |

#### tenant（租户组）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **id** | 唯一自增id | int |
| **title** | 客户组名称 | int |
| **address** | 地址 | varchar |
| **email** | 邮箱 | varcahr |
| **phone** | 手机号 | varcahr |
| **additional\_info** | 备注 | varchar |

#### tenants（3D平台租户）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **id** | 唯一自增id | int |
| **sceneModelNum** | 场景模型数 | int |
| **deviceModelNum** | 设备模型数 | int |
| **createdAt** | 创建时间 | bigint |
| **updatedAt** | 更新时间 | bigint |
| **version** | 版本 | bigint |

#### customer（客户组）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **id** | 唯一自增id | int |
| **tenant\_id** | 租户id | int |
| **title** | 客户组名称 | int |
| **address** | 地址 | varchar |
| **email** | 邮箱 | varcahr |
| **phone** | 手机号 | varcahr |
| **additional\_info** | 备注 | varchar |

#### device(设备)

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **id** | 设备Id | UUID |
| **tenant\_Id** | 租户Id | int |
| **customer\_Id** | 客户Id | int |
| **name** | 设备名称 | text |
| **additional\_info** | 附加信息 | text |
| **manufacture** | 设备类型 | text |
| **device\_type** | 厂商 | text |
| **model** | 型号 | text |
| **location** | 设备位置 | text |
| **Status** | 设备状态 | tinyint |
| **Search\_text** | 搜索框 | text |
| **parent\_device\_id** | 父设备Id | text |
| **site\_id** | 站点Id | text |
| **life\_time** | 维修年限 | bigint |

#### devicesModel（设备模型）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **id** | 设备模型Id | int |
| **tenant\_id** | 租户Id | int |
| **device\_id** | 设备Id | varchar |
| **name** | 设备名称 | varchar |
| **deviceModelUrl** | 设备模型url | varchar |
| **compressStatus** | 压缩状态 | tinyint |
| **ossStatus** | OSS状态 | tinyint |
| **label** | 标签 | text |
| **location** | 设备位置 | varchar |
| **createAt** | 创建时间 | bigint |
| **updatedAt** | 更新时间 | bigint |
| **version** | 版本 | bigint |
| **siteId** | 站点Id | int |

#### device\_credentials（设备认证）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **id** | Id | UUID |
| **device\_Id** | 设备id | UUID |
| **tocken** | 设备访问令牌 | text |

#### group（设备组）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **id** | 设备组ID号 | UUID |
| **tenant\_id** | 租户ID号 | int |
| **customer\_id** | 用户ID号 | int |
| **name** | 设备组名称 | text |

#### device\_by\_group\_id（设备-设备组对应关系）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **device\_id** | 设备ID号 | UUID |
| **group\_id** | 设备组ID号 | UUID |

#### rule（报警规则）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **ruleId** | 规则ID | int |
| **tenantId** | 规则所属租户ID | int |
| **addition\_info** | 备注信息 | varchar |
| **name** | 规则名 | varchar |
| **state** | 规则当前状态 | varchar |

#### Filter（过滤器）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **filterId** | 过滤器ID | int |
| **Type** | 过滤器类型 | varchar |
| **jsCode** | Js代码 | varchar |
| **name** | 过滤器名称 | varchar |

#### rule2Filter\_Relation（规则与过滤器对应关系）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **filterId** | 过滤器ID | int |
| **ruleId** | 规则ID | int |

#### transform（插件）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **transformId** | 插件的ID | int |
| **requestBody** | 请求体 | varchar |
| **url** | 请求的URL | varchar |
| **Name** | 插件名称 | varchar |
| **Method** | 请求方法 | varchar |

#### rule2TransForm\_Relation（规则与过滤器对应关系）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **filterId** | 过滤器ID | int |
| **transformId** | 插件ID | int |

#### ability（能力）

厂商：

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **manufacturer\_id** | 唯一自增Id | int |
| **manufacturer\_name** | 厂商名称 | varchar |

设备类型：

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **deviceType\_id** | 唯一自增Id | int |
| **manufacturer\_id** | 设备i类型所属厂商Id | int |
| **deviceType\_name** | 设备类型名称 | varchar |

型号：

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **model\_id** | 唯一自增Id | int |
| **deviceType\_id** | 型号所属设备Id | int |
| **manufacturer\_id** | 设备类型所属厂商Id | int |
| **model\_name** | 型号名称 | varchar |

服务：

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **service\_id** | 唯一自增Id | int |
| **model\_id** | 服务组Id | int |
| **service\_des** | 服务描述 | varchar |

#### attributes\_kv\_cf（属性数据）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **entity\_Id** | 设备Id | UUID |
| **last\_update\_ts** | 时间戳 | bigint |
| **attribute\_key** | 键（数据名称） | text |
| **bool\_v** | BOOLEAN数据值 | boolean |
| **str\_v** | STRING数据值 | text |
| **long\_v** | LONG数据值 | bigint |
| **dbl\_v** | DOUBLE数据值 | double |

#### ts\_kv\_cf（遥测数据）

ts\_kv\_cf:

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **device\_Id** | 设备Id | UUID |
| **ts** | 时间戳 | bigint |
| **key** | 键（数据名称） | text |
| **bool\_v** | BOOLEAN数据值 | boolean |
| **str\_v** | STRING数据值 | text |
| **long\_v** | LONG数据值 | bigint |
| **dbl\_v** | DOUBLE数据值 | double |
| **partition** | 数据分区 | bigint |

ts\_kv\_partitions\_cf（存储数据的分区信息）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **entity\_Id** | 设备Id | UUID |
| **key** | 键（数据名称） | text |
| **partition** | 数据分区 | bigint |

ts\_kv\_latest\_cf（存储数据的最新数据）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **device\_Id** | 设备Id | UUID |
| **ts** | 时间戳 | bigint |
| **key** | 键（数据名称） | text |
| **bool\_v** | BOOLEAN数据值 | boolean |
| **str\_v** | STRING数据值 | text |
| **long\_v** | LONG数据值 | bigint |
| **dbl\_v** | DOUBLE数据值 | double |

#### event（事件）：

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **id** | 事件Id | UUID |
| **tenant\_id** | 租户ID | int |
| **event\_type** | 事件类型 | text |
| **entity\_type** | 实体类型（例如：设备） | text |
| **entity\_id** | 实体ID（例如：设备ID） | text |
| **body** | 事件体 | text |

#### data\_model（数据模型）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **model\_id** | 唯一自增Id | int |
| **model\_desc** | 模型描述 | varchar |
| **model\_input** | 模型输入 | varchar |
| **model\_name** | 模型名称 | varchar |
| **model\_path** | 模型路径 | varchar |
| **tenant\_id** | 模型所属租户Id | int |

#### update\_message（更新消息）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **id** | 唯一自增Id | int |
| **tenant\_id** | 所属租户Id | int |
| **message** | 消息内容 | varchar |
| **messageType** | 消息类型 | varchar |
| **ts** | 时间戳 | bigint |

#### dashboard（仪表盘）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **id** | 唯一自增Id | int |
| **tenant\_id** | 所属租户Id | int |
| **customer\_id** | 所属客户Id | int |
| **name** | 名称 | varchar |

#### widget\_entity（部件实体）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **id** | 唯一自增Id | int |
| **dashboard\_id** | 所属仪表盘Id | int |
| **device\_id** | 绑定设备ID | varchar |
| **name** | 名称 | varchar |
| **entity\_type** | 实体类型 | varchar |

#### sites（站点）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **id** | 站点Id | int |
| **tenant\_id** | 租户Id | int |
| **name** | 站点名称 | varchar |
| **sceneUrl** | 场景url | varchar |
| **compressStatus** | 压缩状态 | tinyint |
| **ossStatus** | OSS状态 | tinyint |
| **longtitude** | 经度 | double |
| **latitude** | 纬度 | double |
| **createAt** | 创建时间 | bigint |
| **updatedAt** | 更新时间 | bigint |
| **version** | 版本 | bigint |
| **devicesModelCount** | 设备模型数 | int |
| **sceneModelLoca** | 场景模型位置 | varchar |

#### recent\_data（近期数据）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **tenant\_id** | 租户Id | int |
| **device\_type** | 设备类型 | varchar |
| **max\_value** | 最大值 | decimal |
| **min\_value** | 最小值 | decimal |
| **mean\_value** | 均值 | decimal |
| **stddev\_value** | 标准差 | decimal |
| **data\_count** | 数据数 | int |
| **usual\_data\_count** | 正常数据数 | int |
| **usual\_data\_rate** | 正常数据率 | decimal |
| **flag** | flag | varchar |
| **date** | 日期 | date |

#### recent\_device（近期设备）

|  |  |  |
| --- | --- | --- |
| **字段** | **描述** | **数据类型** |
| **tenant\_id** | 租户Id | int |
| **device\_type** | 租户Id | varchar |
| **device\_count** | 设备数 | int |
| **data\_count** | 数据数 | int |
| **usual\_data\_count** | 正常数据数 | int |
| **usual\_data\_rate** | 正常数据率 | decimal |
| **flag** | flag | varchar |
| **date** | 日期 | date |

## 4安全设计

### 4.1 Cassandra

Cassandra数据库提供了三个安全组件：

* 用于客户端和节点间以及节点与节点间通信的TLS/SSL加密
* Client authentication
* Authorization

#### 4.1.1 TSL/SSL加密

Cassandra提供客户端与数据库集群之间以及集群内节点之间的安全通信。 启用加密可确保数据不受影响并可安全传输。客户端到节点和节点到节点加密的选项分开管理，可以独立配置。

管理节点间加密的设置可在cassandra.yaml的server\_encryption\_options部分中找到。要启用节点间加密，请将internode\_encryption设置从默认值none更改为rack，dc或all中的一个值。

管理客户端到节点加密的设置可在cassandra.yaml的client\_encryption\_options部分中找到。 这里有两个主要设置用于启用加密，enabled和optional。

如果两者均未设置为true，则客户端连接完全未加密。

如果enabled设置为true并且optional设置为false，则加密所有客户端连接。

如果两个选项均设置为true，则端口支持加密和未加密连接。这种配置下，使用加密的客户端连接将会被自动检测并由服务器处理。

作为optional设置的替代方案，也可以将单独的端口配置为符合要求的安全和不安全连接。为此，可将optional设置为false，并使用cassandra.yaml中的native\_transport\_port\_ssl设置指定用于客户端安全通信的端口。

#### 4.1.2 Client authentication

Cassandra支持客户端的身份验证，并使用cassandra.yaml中的身份验证设置进行配置。Cassandra提供了两个选项。

默认情况下，Cassandra配置了AllowAllAuthenticator，它不执行身份验证检查，因此不需要凭证。它完全禁用身份验证。

Cassandra还包括PasswordAuthenticator，它将加密的凭证存储在系统表中。 这可以用来启用简单的用户名/密码认证。

#### 4.1.3 Authorization

Cassandra支持客户端的Authorization，并使用cassandra.yaml中的Authorization设置进行配置。 Cassandra提供了两个选项。

默认情况下，Cassandra配置了AllowAllAuthorizer，它不执行任何检查，因此有效地授予角色所有权限。 如果AllowAllAuthenticator是配置的Client authentication选项，则Authorization必须使用AllowAllAuthorizer。

默认分配还包括CassandraAuthorizer，它实现了完整的权限管理功能并将其数据存储在Cassandra系统表中。

### MySQL

#### 4.2.1 安装最新的补丁

在mysql命令行下查询MySQL的版本：

SHOW VARIABLES WHERE Variable\_name LIKE "version";

确认是否由需要安装的补丁包，如果有请安装。

#### 4.2.2 删除test数据库

Mysql数据库默认安装好后，存在一个名为test的数据库，如果存在，请执行如下命令删除：

Drop database “test”

#### 4.2.3 控制可以访问所有数据库的账号

Mysql数据库下的user表和db表中存放着可以授予数据库用户的权限，确保只有管理员账号才能访问所有数据库。可以访问mysql数据库的用户或许可以查看密码哈希值、修改用户权限等等。

使用如下sql语句：

SELECT user, host FROM mysql.user

WHERE (Select\_priv = 'Y') OR (Insert\_priv = 'Y') OR (Update\_priv = 'Y')

OR (Delete\_priv = 'Y') OR (Create\_priv = 'Y') OR (Drop\_priv = 'Y');

SELECT user, host FROM mysql.db WHERE db = 'mysql'

AND ((Select\_priv = 'Y') OR (Insert\_priv = 'Y') OR (Update\_priv = 'Y')

OR (Delete\_priv = 'Y') OR (Create\_priv = 'Y') OR (Drop\_priv = 'Y'));

确保返回结果只能是数据库管理员账号。

#### 4.2.4 限制非管理员用户的权限

Mysql.user表中的权限列有：

file\_priv：表示是否允许用户读取数据库所在主机的本地文件；

Process：表示是否允许用户查询所有用户的命令执行信息；

Super\_priv：表示用户是否有设置全局变量、管理员调试等高级别权限；

Shutdown\_priv：表示用户是否可以关闭数据库；

Create\_user\_priv：表示用户是否可以创建或删除其他用户；

Grant\_priv：表示用户是否可以修改其他用户的权限。

应确保只有数据库管理员才有上述权限，使用如下sql语句查看拥有各个权限的数据库账号：

select user, host from mysql.user where File\_priv = 'Y';

select user, host from mysql.user where Process\_priv = 'Y';

select user, host from mysql.user where Process\_priv = 'Y';

SELECT user, host FROM mysql.user WHERE Shutdown\_priv = 'Y';

SELECT user, host FROM mysql.user WHERE Create\_user\_priv = 'Y';

SELECT user, host FROM mysql.user WHERE Grant\_priv = 'Y'；

SELECT user, host FROM mysql.db WHERE Grant\_priv = 'Y';

确保查询结果中不存在非管理员用户。

如果存在非管理员用户，使用如下命令进行权限回收：

REVOKE FILE ON \*.\* FROM '<user>';

REVOKE PROCESS ON \*.\* FROM '<user>';

REVOKE SUPER ON \*.\* FROM '<user>';

REVOKE SHUTDOWN ON \*.\* FROM '<user>';

REVOKE CREATE USER ON \*.\* FROM '<user>';

REVOKE GRANT OPTION ON \*.\* FROM <user>;

其中user为上述查询到的非管理员用户。