元数据版本控制文档

1. 版本控制目的

能够在实际业务结构发生变化时，在元数据管理系统中可以跟着业务系统的变化，生成不同的业务版本信息。通过版本和版本之间的对比可以看到整个元数据的变更信息。

1. 版本控制表设计
2. 创建mdm\_technology\_column\_[master](https://www.baidu.com/link?url=wrEAC7V9wF98VMv30ysBaYy7V2ywbjUvQCSFTh6Y30AJAVcv54ArK6OK75CwcQIyTJHmNQdxdaBTaX67WZDxHBB-MsToD9q9GEniUAb6LFy&wd=&eqid=86a86ec6000b0e6a000000045de8922b)表结构和mdm\_technology\_column保持完全一致，修改mdm\_technology\_column表名为mdm\_technology\_column\_host。
3. 创建mdm\_technology\_column\_edition字段版本表，表结构在mdm\_technology\_column表的基础上添加 version\_id:版本号唯一标识字段。
4. 修改mdm\_data\_element表名为mdm\_data\_element\_host，创建mdm\_data\_element\_master表，表结构和mdm\_data\_element保持一致。
5. 创建mdm\_data\_element\_edition：数据元版本表，表结构在mdm\_data\_element的基础上添加version\_id:版本号唯一标识字段。
6. 创建mdm\_edition\_info元数据版本信息表：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段中文 | 类型 | 描述 |
| id | 自增长id | Int |  |
| version\_code | 版本编号 | Varchar(50) | MDM\_yyyyddmmhhmmss\_tableId;以\_分段 MDM默认前段，中间段为生成版本的时间到秒,后一段为生成版本表的唯一标识 |
| description | 版本描述 | Varchar(800) | 对当前生成版本的描述信息 |
| user\_account | 用户账号 | Varchar(20) | 创建人账号 |
| user\_id | 用户唯一标识 | Varchar(50) | 唯一标识 |
| Create\_date\_time | 创建时间 | DateTime | 版本生成时间 |

1. 版本控制方式

技术元数据版本控制以表为单位；业务元数据版本控制以数据集为单位。

1. 元数据管理系统中原有的功能不变，全局搜索配置Mapper.xml文件中的包含mdm\_technology\_column，mdm\_data\_element所有的sql在这两个表名后统一添加\_{edition}占位符。
2. 元数据系统自身访问后台服务打包时默认把edition占位符替换为host,元数据所有操作数据都放到对应host后缀的表中。
3. 其它应用系统调用资产目录访问，后台服务打包默认把edition占位服务替换为master,应用只获取当前已经启动的版本资产目录，在元数据中编辑且未发布的版本不影响应用系统正常使用。
4. 元数据在技术视图中，添加了库表信息正常添加，添加字段默认只对元数据系统可见，其它应用只有在元数据系统发布版本后才能正常使用。
5. 元数据发布版本首先需要把其它应用系统，现在使用的数据从mdm\_data\_element\_master，mdm\_data\_element\_master表中复制到mdm\_technology\_column\_edition, mdm\_data\_element\_edition表中，并且在发布版本时需要用户添加版本描述，需要把版本描述添加到mdm\_edition\_info表中并且根据规则生成版本编码。把生成版本的自增长id，赋值到mdm\_technology\_column\_edition, mdm\_data\_element\_edition表中对应数据的version\_id字段中。
6. 生成新版本后，把当前mdm\_technology\_column\_master, mdm\_data\_element\_master删除，然后把mdm\_technology\_column\_host, mdm\_data\_element\_host表中的所有数据复制到mdm\_technology\_column\_master, mdm\_data\_elementmaster表中。包括自增长id都必须保持一致。
7. 版本回滚，回滚当前应用系统使用的版本，需要把mdm\_technology\_column\_host, mdm\_data\_element\_host清空，然后把mdm\_technology\_column\_master, mdm\_data\_element\_master复制过来，并且保证自增长id一致。历史版本回滚同理。
8. 版本维护

版本对比功能：

1. 在技术视图、业务视图树形菜单中添加版本展示按钮，点击版本展示按钮，查询版本表信息，并把当前所有的版本信息进行展示，版本信息表中默认添加一条当前应用使用版本版本编号为”MDM\_DEFAULT”，如果双击版本信息默认展示元数据系统当前数据信息和双击的版本数据进行版本对比。 如果勾选了两个版本信息点击比较，那么默认对当前两个历史版本的数据信息进行对比。历史版本比较根据version\_id区分比较数据。版本信息展示页面，复选框最多只能选择两个进行比较。
2. 版本信息展示页面默认用表格形式展示，展示信息包含版本编号，版本描述，创建时间，创建人账号。
3. 当前元数据管理系统和应用系统元数据比较，如果没有选择比较的元数据默认整个技术视图或者业务视图进行对比，对比时默认左侧展示技术元数据或者业务元数据的树行菜单。技术元数据点击表，中间展示应用系统下的所有字段信息，右侧展示元数据系统对应表中的字段信息，并且标注两个版本中间的差异。
4. 业务视图同理，右侧展示数据集，点击数据集中间展示应用系统的属性信息，右侧展示元数据系统属性信息。是对（mdm\_data\_element\_master，mdm\_data\_element\_master和mdm\_data\_element\_host，mdm\_data\_element\_host）表里面的数据对比。
5. 如果是应用版本和历史版本对比，那么对比的数据来源为mdm\_data\_element\_master，mdm\_data\_element\_master和mdm\_technology\_column\_edition, mdm\_data\_element\_edition查询版本表需要根据version\_id查询对应版本。
6. 如果两个历史记录进行对比，数据来源都是mdm\_technology\_column\_edition, mdm\_data\_element\_edition表，查询对比信息需要根据当前表中的version\_id来区分对比信息。

**发布版本**

DELIMITER //

DROP PROCEDURE IF EXISTS releaseVersion;

CREATE PROCEDURE releaseVersion(

in versionId INTEGER

)

BEGIN

DECLARE result\_code INTEGER DEFAULT 0;

DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET result\_code=1;

START TRANSACTION; -- 开始事务

insert into mdm\_technology\_column\_edition

SELECT id,table\_id,resource\_id,resource\_code,name\_en,name\_cn,data\_type,data\_length,

data\_type\_group,null\_allow,primary\_key,comment,versionId as version\_id

FROM mdm\_technology\_column\_master;

insert into mdm\_data\_element\_edition

select id,resource\_id,resource\_code,source\_id,category,name\_cn,name\_en,definition,data\_type, show\_format,allowable\_value,

allowable\_type,allowable\_code,versionId as version\_id

from mdm\_data\_element\_master;

delete from mdm\_technology\_column\_master;

delete from mdm\_data\_element\_master;

insert into mdm\_technology\_column\_master

select id,table\_id,resource\_id,resource\_code,name\_en,name\_cn,data\_type,data\_length,

data\_type\_group,null\_allow,primary\_key,comment FROM mdm\_technology\_column\_host;

insert into mdm\_data\_element\_master

select id,resource\_id,resource\_code,source\_id,category,name\_cn,name\_en,definition,data\_type ,show\_format,allowable\_value,allowable\_type,allowable\_code from mdm\_data\_element\_host;

IF result\_code = 1 THEN

ROLLBACK;

ELSE

COMMIT;

END IF;

select result\_code;

END;

**回滚master**

DELIMITER //

DROP PROCEDURE IF EXISTS rollbackMasterVersion;

CREATE PROCEDURE rollbackMasterVersion()

BEGIN

DECLARE result\_code INTEGER DEFAULT 0;

DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET result\_code=1;

START TRANSACTION; -- 开始事务

delete from mdm\_technology\_column\_host;

delete from mdm\_data\_element\_host;

insert into mdm\_technology\_column\_host

SELECT id,table\_id,resource\_id,resource\_code,name\_en,name\_cn,data\_type,data\_length,

data\_type\_group,null\_allow,primary\_key,comment FROM mdm\_technology\_column\_master;

insert into mdm\_data\_element\_host

select id,resource\_id,resource\_code,source\_id,category,name\_cn,name\_en,definition,data\_type, show\_format,allowable\_value,allowable\_type,allowable\_code

from mdm\_data\_element\_master;

IF result\_code = 1 THEN

ROLLBACK;

ELSE

COMMIT;

END IF;

select result\_code;

END;

**回滚edition**

DELIMITER //

DROP PROCEDURE IF EXISTS rollbackEditionVersion;

CREATE PROCEDURE rollbackEditionVersion(

in versionId INTEGER

)

BEGIN

DECLARE result\_code INTEGER DEFAULT 0;

DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET result\_code=1;

START TRANSACTION; -- 开始事务

delete from mdm\_technology\_column\_host;

delete from mdm\_data\_element\_host;

insert into mdm\_technology\_column\_host

SELECT id,table\_id,resource\_id,resource\_code,name\_en,name\_cn,data\_type,data\_length,

data\_type\_group,null\_allow,primary\_key,comment

FROM mdm\_technology\_column\_edition where version\_id = versionId;

insert into mdm\_data\_element\_host

select id,resource\_id,resource\_code,source\_id,category,name\_cn,name\_en,definition,data\_type, show\_format,allowable\_value,allowable\_type,allowable\_code

from mdm\_data\_element\_edition where version\_id = versionId;

IF result\_code = 1 THEN

ROLLBACK;

ELSE

COMMIT;

END IF;

select result\_code;

END;