# 拉链表实现对历史数据的存储

## 背景

为了记录数据的历史状态，确保追溯数据的历史信息，保证小孩生成报告的数据不会被篡改，对于可以被修改或者被删除的数据，通过拉链表的方式，以天为单位记录数据的变化。

## 原理

## 概念

拉链表是针对数据仓库设计中表存储数据的方式而定义的，顾名思义，所谓拉链，就是记录历史。记录一个事物从开始，一直到当前状态的所有变化的信息。

## 技术实现

## 数据结构

拉链表与拉链来源表的结构相比，增加了三列：

* 自增id：拉链表id，可以没有；
* 开始日期：数据的产生时间。比如数据是在2月1日添加的，2月2日凌晨运行程序时，填入”2020-2-2”
* 结束日期：当前有效的数据，结束日期填写“2999-12-31”，对于已经删除或者修改的数据，填写删除或者修改动作发生日期的第二日。

## 实现

## 范围

针对入库后可能会被修改或被删除的数据

* **学校**
* 学校信息表（contents\_center）
* 学校元数据表（centers\_metadata）
* **班级**
* 班级信息表（contents\_group）
* 班级元数据表（groups\_metadata），
* 班级周期表（groups\_period）
* **小孩**
* 小孩信息表（contents\_enrollment）
* 小孩自定义属性表（enrollments\_metedata）,包含评分的完成时间,drdp属性信息等
* 小孩周期表（enrollment\_period），包含使用的框架
* **观察**
* 观察表（notes\_note）
* **评分**
* 评分表（contents\_enrollment\_score）
* 观察小孩关系表（notes\_noteenrollment）
* 观察媒体关系表（notes\_notemedia），保存观察中对应的视频、图片信息
* 模版/测评点表（domains\_domain），测评点码表
* 观察测评点关系表（notes\_notedomain），记录观察时选择的测评点

## 命名规范

1. 原表前面增加”ods\_”，如contents\_enrollment=》ods\_contents\_enrollment
2. 表内增加data\_id(*uniqueidentifier*)、data\_start\_date(datetime)、data\_end\_date(datetime)

## 单张表的实现模板

以contents\_enrollment表为例，程序部署日期假设为2020-3-1，程序执行时“2020-3-1”是入参

1. **建表**

*--拉链表*

*CREATE TABLE ods\_contents\_enrollment (*

*Id uniqueidentifier NOT NULL DEFAULT (newsequentialid()) ,*

*DisplayName nvarchar(128) NOT NULL ,*

*IsActived bit NOT NULL ,*

*IsGraduated bit NOT NULL DEFAULT ((0)) ,*

*IsDeleted bit NOT NULL DEFAULT ((0)) ,*

*CreateAtUtc datetime NOT NULL ,*

*AvatarMediaId uniqueidentifier NULL ,*

*GroupId uniqueidentifier NOT NULL ,*

*FirstName nvarchar(128) NULL ,*

*LastName nvarchar(128) NULL ,*

*ArchivedGroupId uniqueidentifier NULL ,*

*ArchivedAtUtc datetime NULL ,*

*MiddleName nvarchar(128) NULL ,*

*BirthDate date NULL ,*

*Gender varchar(8) NULL ,*

*EnrollmentDate date NULL ,*

*WithdrawnDate date NULL ,*

*PrivatePhoto bit NOT NULL DEFAULT ((0)) ,*

*IsInactive bit NOT NULL DEFAULT ((0)) ,*

*SourceGroupId uniqueidentifier NULL ,*

*Version datetime NOT NULL DEFAULT (getutcdate()) ,*

*Email nvarchar(256) NULL ,*

*PhoneNumber nvarchar(256) NULL ,*

*data\_id uniqueidentifier not null DEFAULT (newsequentialid()),*

*data\_start\_date datetime NULL ,*

*data\_end\_date datetime NULL ,*

*CONSTRAINT pk\_ods\_contents\_enrollment PRIMARY KEY (data\_id)*

*);*

*create index idx1\_ods\_contents\_enrollment on ods\_contents\_enrollment(data\_end\_date,id);*

*--辅助提高效率的临时表*

*CREATE TABLE ods\_contents\_enrollment\_tmp1 (*

*Id uniqueidentifier NOT NULL DEFAULT (newsequentialid()) ,*

*data\_id uniqueidentifier not null DEFAULT (newsequentialid()),*

*hashvalue varchar(100) ,*

*CONSTRAINT pk\_ods\_contents\_enrollment\_tmp1 PRIMARY KEY (id)*

*);*

*CREATE TABLE ods\_contents\_enrollment\_tmp2 (*

*Id uniqueidentifier NOT NULL DEFAULT (newsequentialid()) ,*

*hashvalue varchar(100) ,*

*CONSTRAINT PK\_ods\_contents\_enrollment\_tmp2 PRIMARY KEY (id)*

*);*

*CREATE TABLE ods\_contents\_enrollment\_tmp3 (*

*Id uniqueidentifier NOT NULL DEFAULT (newsequentialid()) ,*

*data\_id uniqueidentifier ,*

*flg varchar(100) ,*

*CONSTRAINT PK\_ods\_contents\_enrollment\_tmp3 PRIMARY KEY (id)*

*);*

1. **初次初始化数据**

*insert into ods\_contents\_enrollment*

*(Id,DisplayName,IsActived,IsGraduated,IsDeleted,*

*CreateAtUtc,AvatarMediaId,GroupId,FirstName,LastName,*

*ArchivedGroupId,ArchivedAtUtc,MiddleName,BirthDate,Gender,*

*EnrollmentDate,WithdrawnDate,PrivatePhoto,IsInactive,SourceGroupId,*

*Version,Email,PhoneNumber,data\_start\_date,data\_end\_date)*

*select*

*Id,DisplayName,IsActived,IsGraduated,IsDeleted,*

*CreateAtUtc,AvatarMediaId,GroupId,FirstName,LastName,*

*ArchivedGroupId,ArchivedAtUtc,MiddleName,BirthDate,Gender,*

*EnrollmentDate,WithdrawnDate,PrivatePhoto,IsInactive,SourceGroupId,*

*Version,Email,PhoneNumber,convert(datetime,'2020-3-1', 20),convert(datetime,'2999-12-31', 20)*

*from contents\_enrollment;*

1. **后续插入数据**
   1. **读取拉链表中当前有效数据，放入临时表，生成hash值**

**// cast和convert将某种数据类型的表达式显式准换为另一种数据类型。**

**// cast(expression as data\_type) : expression:任何有效的SQLSERVER表达式，as:用于分割两个参数，在as前是要处理的数据，as后是要转换的数据类型。data\_type:目标系统中所提供的数据类型**nchar、nvarchar、char、varchar、binary 或 varbinary 数据类型 如：select cast(‘12’ AS int)

**// convert(data\_type[(length)],expression[,style])**

**// data\_type:目标系统中所提供的数据类型**nchar、nvarchar、char、varchar、binary 或 varbinary 数据类型的**。Length是指定数据类型长度的可选整数，默认30。Style也是可选参数。**

**Style:** 日期格式样式，借以将 datetime 或 smalldatetime 数据转换为字符数据（nchar、nvarchar、char、varchar、nchar 或 nvarchar 数据类型）；或者字符串格式样式，借以将 float、real、money 或 smallmoney 数据转换为字符数据（nchar、nvarchar、char、varchar、nchar 或 nvarchar 数据类型）。

*truncate table ods\_contents\_enrollment\_tmp1;*

*insert into ods\_contents\_enrollment\_tmp1*

*(Id,data\_id,hashvalue)*

*select*

*Id,data\_id,*

*CONVERT(VARCHAR(32), HashBytes('MD5',*

*isnull(cast(Id as varchar(100)),'')+isnull(cast(DisplayName as varchar(100)),'')+isnull(cast(IsActived as varchar(100)),'')+isnull(cast(IsGraduated as varchar(100)),'')+isnull(cast(IsDeleted as varchar(100)),'')+isnull(cast(*

*CreateAtUtc as varchar(100)),'')+isnull(cast(AvatarMediaId as varchar(100)),'')+isnull(cast(GroupId as varchar(100)),'')+isnull(cast(FirstName as varchar(100)),'')+isnull(cast(LastName as varchar(100)),'')+isnull(cast(*

*ArchivedGroupId as varchar(100)),'')+isnull(cast(ArchivedAtUtc as varchar(100)),'')+isnull(cast(MiddleName as varchar(100)),'')+isnull(cast(BirthDate as varchar(100)),'')+isnull(cast(Gender as varchar(100)),'')+isnull(cast(*

*EnrollmentDate as varchar(100)),'')+isnull(cast(WithdrawnDate as varchar(100)),'')+isnull(cast(PrivatePhoto as varchar(100)),'')+isnull(cast(IsInactive as varchar(100)),'')+isnull(cast(SourceGroupId as varchar(100)),'')+isnull(cast(*

*Version as varchar(100)),'')+isnull(cast(Email as varchar(100)),'')+isnull(cast(PhoneNumber as varchar(100)),'')*

*), 2) hashvalue*

*from ods\_contents\_enrollment*

*where data\_end\_date=convert(datetime,'2999-12-31', 20) ;*

* 1. **整理来源表数据，生成hash值**

*truncate table ods\_contents\_enrollment\_tmp2;*

*insert into ods\_contents\_enrollment\_tmp2*

*(Id,hashvalue)*

*select*

*Id,*

*CONVERT(VARCHAR(32), HashBytes('MD5',*

*isnull(cast(Id as varchar(100)),'')+isnull(cast(DisplayName as varchar(100)),'')+isnull(cast(IsActived as varchar(100)),'')+isnull(cast(IsGraduated as varchar(100)),'')+isnull(cast(IsDeleted as varchar(100)),'')+isnull(cast(*

*CreateAtUtc as varchar(100)),'')+isnull(cast(AvatarMediaId as varchar(100)),'')+isnull(cast(GroupId as varchar(100)),'')+isnull(cast(FirstName as varchar(100)),'')+isnull(cast(LastName as varchar(100)),'')+isnull(cast(*

*ArchivedGroupId as varchar(100)),'')+isnull(cast(ArchivedAtUtc as varchar(100)),'')+isnull(cast(MiddleName as varchar(100)),'')+isnull(cast(BirthDate as varchar(100)),'')+isnull(cast(Gender as varchar(100)),'')+isnull(cast(*

*EnrollmentDate as varchar(100)),'')+isnull(cast(WithdrawnDate as varchar(100)),'')+isnull(cast(PrivatePhoto as varchar(100)),'')+isnull(cast(IsInactive as varchar(100)),'')+isnull(cast(SourceGroupId as varchar(100)),'')+isnull(cast(*

*Version as varchar(100)),'')+isnull(cast(Email as varchar(100)),'')+isnull(cast(PhoneNumber as varchar(100)),'')*

*), 2) hashvalue*

*from contents\_enrollment;*

* 1. **数据比较，形成比对结果，去掉不变的数据**

*truncate table ods\_contents\_enrollment\_tmp3;*

*insert into ods\_contents\_enrollment\_tmp3*

*(Id,data\_id,flg)*

*SELECT*

*isnull(a.id,b.id) id,a.data\_id,*

*case*

*when a.id is null then 'new'*

*when b.id is null then 'deleted'*

*else 'modified'*

*end flg*

*from ods\_contents\_enrollment\_tmp1 a full join ods\_contents\_enrollment\_tmp2 b on a.id=b.id*

*where isnull(a.hashvalue,'1')<>isnull(b.hashvalue,'0');*

* 1. **修改删除数据和修改数据的结束日期**

*update a set a.data\_end\_date=convert(datetime,'2020-3-1', 20)-1*

*from ods\_contents\_enrollment a ,ods\_contents\_enrollment\_tmp3 b*

*where a.data\_id=b.data\_id and b.flg in('deleted','modified');*

* 1. **插入新增数据、修改数据，插入新数据开始日期是本日**

*insert into ods\_contents\_enrollment*

*(Id,DisplayName,IsActived,IsGraduated,IsDeleted,*

*CreateAtUtc,AvatarMediaId,GroupId,FirstName,LastName,*

*ArchivedGroupId,ArchivedAtUtc,MiddleName,BirthDate,Gender,*

*EnrollmentDate,WithdrawnDate,PrivatePhoto,IsInactive,SourceGroupId,*

*Version,Email,PhoneNumber,data\_start\_date,data\_end\_date)*

*select*

*a.Id,DisplayName,IsActived,IsGraduated,IsDeleted,*

*CreateAtUtc,AvatarMediaId,GroupId,FirstName,LastName,*

*ArchivedGroupId,ArchivedAtUtc,MiddleName,BirthDate,Gender,*

*EnrollmentDate,WithdrawnDate,PrivatePhoto,IsInactive,SourceGroupId,*

*Version,Email,PhoneNumber,convert(datetime,'2020-3-1', 20),convert(datetime,'2999-12-31', 20)*

*from contents\_enrollment a,ods\_contents\_enrollment\_tmp3 b*

*where a.id=b.id and b.flg in('new','modified');*

## Lamda实现

1. 获取当前日期，10位字符串yyyy-mm-dd（0点以后执行）
2. 将日期字符串作为参数传给各脚本执行
3. 记录拉链表运行日志，开始时记录start\_or\_end='start'的一条记录，结束时记录start\_or\_end='end'的一条记录，同时记录状态：success、failue
4. 注意单个脚本的执行时间应远小于15分钟。

## 数据记录

1. 小孩锁定时，记录小孩的锁定时间，报告名称等信息

表结构：记录评分的完成时间、周期锁定时间，周期名、报告名

1. 拉链表数据同步运行日志

拉链表运行日志表(dw\_sys\_etl\_record)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Code | Default Value | Data Type | Primary | Foreign Key | Mandatory |
| Id | Id | newsequentialid() | uniqueidentifier | TRUE | FALSE | TRUE |
| 业务日期 | data\_date |  | varchar(10) | FALSE | FALSE | TRUE |
| 拉链表表名 | taget\_table |  | varchar(100) | FALSE | FALSE | TRUE |
| 步骤编号 | step\_code |  | varchar(100) | FALSE | FALSE | FALSE |
| 步骤描述 | step\_desc |  | varchar(300) | FALSE | FALSE | TRUE |
| 开始/结束 | start\_or\_end |  | varchar(10) | FALSE | FALSE | TRUE |
| 完成时间 | record\_time |  | datetime | FALSE | FALSE | TRUE |
| 运行状态 | status |  | varchar(10) | FALSE | FALSE | FALSE |

create table dw\_sys\_etl\_record

(

Id uniqueidentifier NOT NULL DEFAULT (newsequentialid()),

data\_date varchar(10) not null,

taget\_table varchar(100) not null,

step\_code varchar(100),

step\_desc varchar(300) not null,

start\_or\_end varchar(10) not null,

record\_time datetime not null,

status varchar(10),

CONSTRAINT pk\_dw\_sys\_etl\_record PRIMARY KEY (id)

);

## 数据使用

* 场景一、历史时间点的报告

比如2019-11-2用户点击归档，在数据库中记录该小孩该周期的归档日期2019-11-2，2019-11-3数据进入拉链表。查询时，每张拉链表的数据过滤方式：where data\_start\_date<=2019-11-3 and 2019-11-3<data\_end\_date

其他逻辑不变

* 场景二、生成历史统计数据

计算2019-11-2这一天的学校工作统计情况，同样每张拉链表的数据过滤方式：where data\_start\_date<=2019-11-3 and 2019-11-3<data\_end\_date

其他逻辑不变

* 场景三、小孩档案