

基于起源标注的数据中台原始库数据溯源研究

周学文, 薛 猛

(中国人民解放军 31306 部队, 四川 成都 610036)

摘 要: 数据中台数据质量审验和问题诊断经常需要追溯指定数据的历史变化。着眼于追溯数据中台源信息系统元数据和数据记录变化, 进行了数据表元数据起源标注和数据表数据记录起源标注设计, 基于数据中台结构化数据在线抽取功能同步生成相应的数据起源标注, 并针对数据表元数据版本变化历史追溯、数据表数据记录版本变化历史追溯和指定时间段树形表数据历史追溯等典型溯源需求给出了具体解决方案。本方案在某部数据中台系统中进行了具体实现, 为数据中台数据生命周期溯源提供了重要支撑。

关键词: 数据起源; 起源标注; 数据中台; 数据版本; 全量抽取; 哈希比对

中图分类号: TP391.3 **文献标识码:** A **DOI:** 10.19358/j.issn.2097-1788.2025.12.007

引用格式: 周学文, 薛猛. 基于起源标注的数据中台原始库数据溯源研究 [J]. 网络安全与数据治理, 2025, 44 (12): 48-54.

Research on data traceability of the original library in data middle platform based on origin annotation

Zhou Xuewen, Xue Meng

(Unit 31306 of PLA, Chengdu 610036, China)

Abstract: Data quality inspection and problem diagnosis often require tracing historical changes in specified data. This paper focuses on the changes of metadata and data records of data source information system, designs the origin annotations of data table metadata and data table data records, generates corresponding data origin annotation based on the structured data online extraction function, and provides specific solutions for the typical traceability requirements of data table metadata version change history, data table data record version change history and data history tracing of tree table data at the specified time. This scheme is specifically implemented in a certain data middle platform system, which provides an important support for the data life cycle traceability of the data middle platform.

Key words: data origin; origin annotation; data middle platform; data version; full extraction; hash alignment

0 引言

由机构编制体制调整带来的机构树变化、由新装备入编或老装备退编所导致的装备分类树变化等基础数据调整是信息系统经常需面对的情况, 如何查询历史数据以及追溯数据变化是一个比较难解决的问题。目前大部分管理信息系统只能通过手工还原历史数据库版本来满足历史数据查询要求, 效率低下且难以满足历史数据自动追溯需求。数据中台建设可对解决此问题提供比较理想的解决方案, 通过结构化数据在线抽取功能, 数据中台支持对能够访问的本地关系型数据进行同构化抽取, 并将数据存储于原始库。在数据抽取过程中, 通过应用数据起源相关技术, 可将所有曾经抽取过的历史数据存

储在历史数据库中, 从而能有效满足历史快照查询、树形表演化历史追踪等数据溯源需求。

本文着眼于追溯数据中台源信息系统元数据和数据记录变化, 分别进行了数据表元数据起源标注和数据表数据记录起源标注设计, 并针对数据表元数据版本变化历史追溯、数据表数据记录版本变化历史追溯和指定时间段机构树演化历史追溯等典型溯源需求给出了具体解决方案。这种追溯类似于零部件拆卸所导致的装备 BOM (Bill of Materials) 历史变化溯源, 文献 [1-2] 针对装备维修阶段的装备 BOM 数据起源追踪, 进行了起源标注的设计和形式化描述, 并给出了装备 BOM 数据起源追溯算法的形式化描述。文献 [1-2] 中的装备维修 BOM 类似于本文中的机构树, 但其追溯只考虑了数据表数据记

录变化,未考虑数据表元数据变化,且其主要针对起源标注设计和起源追溯进行了形式化表示,虽证明了基于起源标注的BOM数据起源追溯可行性,但基本未涉及工程实现细节;本文则提出了基于全量哈希比对的起源标注生成和存储机制,描述了具体工程实现。文献[3]提出了一种数据起源形式化表示模型,其重点是基于形式化表示模型,从多个层面解释全特性SQL和过程语言中的数据起源,主要适用于多表关联聚合情况下数据表元数据的起源追溯;本文重点则在于数据中台在线同构化抽取情况下单条数据表记录和树形表数据记录的起源追溯。文献[4]提出了一种基于时态关系的数据起源模型,利用时态表可以获取关系表在特定时间戳下的历史快照,由于快照会占用较多的存储资源,该文提出了快照的最优放置方案,即计算指定数量的时间戳,使得使用这些时间戳下的快照对查询的优化效果最好。其研究重点在于如何降低时态关系的数据冗余和提高历史快照查询效率,未涉及树形表溯源问题。

1 相关概念及典型问题分析

1.1 相关概念及分析

数据起源研究是当前国际计算机科学领域的研究热点之一,数据起源又叫做数据族系、数据系谱、数据溯源^[5-6],其目的就是研究如何设计、存储起源标注并基于起源标注追踪数据对象的演化历史。对于数据起源目前还没有一个统一的、公认的定义。Woodruff等人^[7]给出了一个最早的数据起源定义:一个数据对象的起源由其整个处理历史组成,包括其出处及所有后续的处理步骤。Buneman等人^[8]认为数据起源用于回溯数据的演进过程,展现数据产生时的来源、数据生命周期的变迁过程以及引起这些变化的因素。2018年,经全国科学技术名词审定委员会审定,数据起源正式成为计算机科学技术名词,在数据库领域指的是追踪和记录数据来源及其在数据库中移动的过程^[9]。

起源标注是运行时存储下来的用于进行数据起源追踪的数据^[5-7],Simmhan等人^[10]直接称之为起源标注元数据。李亚子^[11]分析了数据起源的概念和意义,建立了W7模式的数据起源标注,并比较了逆SQL语句、时序图、有向图和RDF等几种常用的起源标注描述模型。由此可见,数据起源和起源标注是有明显区别的,数据起源是最终展现给用户的数据演化历史视图,起源标注是在数据演化过程中为了日后追踪数据起源而产生的数据。

数据中台源数据主要来源于业务信息系统的关系型数据库,随着业务需求变化,这些源系统数据库的变化主要有两类:一是数据表字段、主键等元数据变化;二

是数据表数据记录发生的变化。针对以上两种变化,均需要生成起源标注,以记录数据表元数据及数据记录的变化历史。该过程相当于为每一个数据表及每一条数据记录创建一个履历表,其主要挑战在于如何以最小的存储代价记录上述变化。另外,机构树、装备分类树、项目分类树等属于一种特殊的数据表,需要对其追溯进行特殊的处理。

1.2 数据表元数据变化溯源需求

数据中台在线数据抽取功能一般会针对每个源系统数据表建立一个数据抽取任务,并支持进行定时或者即时数据抽取。当一个源信息系统数据表字段等元数据发生变化并导致抽取任务不能正常执行时,数据中台系统会提示数据表元数据发生变化并进行元数据比对和矫正,且按规则将原始库中的对应数据表元数据修改为与源系统数据表元数据基本一致,从而满足该数据表后续同构化抽取要求。

数据表元数据变化溯源需求主要有两类:(1)指定时间点,能够还原该时间点原始库指定数据表的数据表结构;(2)指定时间段,能够获取该时间段原始库指定数据表的数据表元数据变化情况。需要针对上述需求,设计数据表元数据起源标注,记录数据表元数据版本并存储相应原始元数据,从而为满足数据表元数据变化溯源需求提供支撑。

1.3 数据表数据记录变化历史溯源需求

数据表数据记录历史追溯是数据中台数据溯源基本需求,主要有三种典型需求:(1)指定数据表某条数据记录,追溯得到指定时间点该数据记录对应的数据记录版本、内容等;(2)指定数据表某条数据记录,追溯得到其在指定时间段内的数据记录变化历史;(3)指定数据表,追溯得到其在指定时间点的历史快照,即该时间点所有数据记录。

数据中台针对每个源系统数据表执行数据同构化抽取任务时,一般有全量抽取和增量抽取两种抽取方式。全量抽取首先将原始库中对应数据表的数据记录清空,然后获取源数据表全部数据记录并全部插入到原始库数据表中,在此过程中不对源表和目标表数据记录进行对比,因此无法知悉哪些数据记录发生了新增、删除和修改等变化。增量抽取是将原始库中数据表数据记录与源系统数据表数据记录进行比对,确定哪些数据记录发生了变化,然后只抽取发生了变化的数据记录。未满足上述数据记录变化历史数据溯源需求时,应该采取增量抽取方式,其主要挑战在于如何以最小代价存储历史数据记录并支撑数据记录快速溯源。

1.4 指定时间段树形表演化历史追溯需求

机构树、装备分类树、项目分类树等属于一种特殊的数据表，一条数据记录代表一个机构（装备分类、项目分类）节点，一般以层次码或者父子关系来标识节点之间的父子关系。由编制体制调整所带来的机构树调整、由新装备入编或老装备退编所导致的装备分类树调整、由项目分类调整所带来的项目分类树调整等是经常存在的客观现象，用到这些基础数据的业务信息系统也不得不据此进行基础数据调整，给相应数据溯源带来了较大挑战。本文以机构树为例，分析树形表溯源需求并给出对应的实现方法。机构树演化历史追溯主要有两种典型追溯需求：一是指定某个机构，追溯得到指定时间点的机构树结构；二是指定某个机构，追溯得到其在指定时间段内的机构树演化历史。

2 基于时间戳的数据表数据起源标注设计

针对数据表元数据和数据记录溯源需求，需要分别设计元数据起源标注和数据记录起源标注，分别记录数据表元数据和数据记录的历史变化。

2.1 数据表元数据起源标注设计

针对数据表元数据溯源需求，在数据中台原始库中建立两个数据表用于记录数据表元数据变化。

(1) 数据表元数据版本管理表，包括数据表 ID、表名、表版本号、开始时间（begintime）、结束时间（endtime）等关键字段，结束时间（endtime）默认为 null，如表 1 所示。

表 1 数据表元数据版本管理表示例

表 ID	表名	表版本号	开始时间 (begintime)	结束时间 (endtime)	表操作 ID	操作人 ID
0001	T	1.0	20240101081326	20240304150607	1	
0002	LYK_T	1.0	20240101081326	20240304150607	1	
0003	T	1.1	20240304150607	null	2	
0004	LYK_T	1.1	20240304050607	null	3	

表 1 中，“表 ID”为自动生成的全局唯一标识符（GUID），示例中以 0001、0002……等表示；“表版本号”从 1.0 开始，每次增加 0.1；“开始时间（begintime）”为该数据表创建时间，“结束时间（endtime）”初始为 null，有字段变化产生时，填写具体变化时间；“表操作 ID”来自于字典表，1 表示初始建表，2 表示添加字段，3 表示修改字段，4 表示删除字段，5 表示修改主键。

(2) 数据表元数据明细管理表，包括数据表 ID、字段 ID、字段名称、字段类型等关键信息，如表 2 所示。

表 2 数据表元数据明细管理表示例

字段 ID	所属数据表 ID	字段名	字段别名	字段类型	精度	字段顺序号	是否主键	来源字段 ID(本表)
1	0001	A	字段 1	varchar	10	1	1	
2	0001	B	字段 2	varchar	20	2	0	
3	0001	C	字段 3	varchar	30	3	0	
4	0002	A	字段 1	varchar	10	1	1	1
5	0002	B	字段 2	varchar	20	2	0	2
6	0002	C	字段 3	varchar	30	3	0	3
7	0003	A	字段 1	varchar	10	1	1	
8	0003	B	字段 2	varchar	20	2	0	
9	0003	C	字段 3	varchar	30	3	0	
10	0003	D	字段 4	varchar	60	4	0	
11	0004	A	字段 1	varchar	10	1	1	7
12	0004	B	字段 2	varchar	20	2	0	8
13	0004	C	字段 3	varchar	30	3	0	9
14	0004	D	字段 4	varchar	60	4	0	

表 2 中，“字段 ID”为自动生成的全局唯一标识符，示例中以 1、2……等表示；“所属数据表 ID”来自于表 1。

针对每一个源信息系统数据表，数据中台在创建对应的原始库数据表（一般取名为“前缀+源数据表名”）时，需要做两步操作：一是在数据表元数据版本管理表中新增一条记录，记录该版本数据表元数据初始版本号、开始时间等信息。如表 1 所示，假设 2024 年 1 月 1 日 8 点 13 分 26 秒（假设数据表中存为 20240101081326）开始创建原始库数据表 T 和 LYK_T，则在元数据版本管理表中创建数据表 T 和 LYK_T 的初始建表标注，表版本号记为 1.0，结束时间（endtime）为默认值 null，表操作类型设为 1，表示初始建表。二是在数据表元数据明细管理表中记录初始版本的数据表字段明细。如表 2 所示，在数据表元数据明细管理表中记录 ID 为“0001”的数据表（对应于表 T_1.0 版本）包含 3 个字段 A、B、C，同时记录其顺序号、主键等关键信息。

假设数据中台在 2024 年 3 月 4 日抽取数据表数据时，发现数据表 T 对应的源系统表元数据发生了变化，导致数据抽取任务未能正常执行，在进行元数据比对并确认源表发生新增字段变化后，则需将现有原始库数据表 T 改名为 T_当前版本号（T_1.0），并参照最新源系统表结构创建一个同名新表 T，同时要进两类操作：

(1) 在元数据版本管理表中记录数据表元数据版本变化。如表 1 所示，进行两步操作：一是将初始 1.0 版本数据表 T 的 endtime 字段填写上修改时间“20240304150607”；二是新增一条标注记录，记录表名为 T，表版本号为 1.1，开始时间（begintime）为“20240304150607”，结束

时间 (endtime) 为默认值 null, 操作类型为 2 (新增字段), 表示表 T 于 2024 年 3 月 4 日 15 点 6 分 7 秒发生了元数据变化并生成了 1.1 新版本。

(2) 在数据表元数据明细管理表中记录相应元数据具体变化。如表 2 所示, 在数据表元数据明细管理表中记录 ID 为“0003”的数据表 (对应于表 T_1.1 版本) 包含 4 个字段 A、B、C、D, 其中 D 为新增字段。

2.2 数据记录数据起源标注设计

为满足数据表数据记录溯源需求, 需针对每一个源

系统数据表, 在数据中台原始库中建立对应的两个数据表用于记录数据表数据记录变化。

数据表数据记录版本管理表, 表名可取为原始表名_log, 包括数据记录 ID (以数据记录对应的哈希值 HASHID 来进行唯一标识)、开始时间 (begintime)、结束时间 (endtime)、操作类型、数据版本号 (以该条记录抽取时间来标识, 一般为该条记录对应的数据抽取任务开始执行时间) 和表元数据版本号等关键字段, 初次生成时结束时间 (endtime) 为默认值 null, 如图 1 所示。

第1次操作	HASHID (HASH主键)	begintime (开始时间)	endtime (结束时间)	operation_type (操作类型)	rele_HASHID (关联记录ID)	D_Version (数据版本号)	T_Version (元数据版本号)
	000001	20240102060708	20240302010101	I	null	20240102060708	1.0
第2次操作	HASHID (HASH主键)	begintime (开始时间)	endtime (结束时间)	operation_type (操作类型)	rele_HASHID (关联记录ID)	D_Version (数据版本号)	T_Version (元数据版本号)
	000001	20240102060708	20240302010101	I	null	20240102060708	1.0
	000002	20240302010101	null	U	000001	20240102060708-001	1.0
第3次操作	HASHID (HASH主键)	begintime (开始时间)	endtime (结束时间)	operation_type (操作类型)	rele_HASHID (关联记录ID)	D_Version (数据版本号)	T_Version (元数据版本号)
	000001	20240102060708	20240302010101	I	null	20240102060708	1.0
	000002	20240302010101	20240306010101	U	000001	20240102060708-001	1.0
	000002	20240306010101	20240306010101	D	000002	20240102060708-002	1.0

图 1 数据表数据记录版本管理表示例

图 1 中, “HASHID” 为根据数据记录内容自动生成的全字段哈希值, 示例中以 000001、000002……等表示; “操作类型” 来自于字典表, I 表示新增记录, U 表示修改记录, D 表示删除记录。当删除记录时, 在数据表数据记录版本管理表中新增一条记录, 由于原始数据记录内容未发生变化, 故 “HASHID” 字段值不变, “operation_type” 设为 D。

针对每一个原始数据表, 在历史库中都要创建一个对应的数据表数据记录历史明细管理表, 用于记录每次抽取时新增或者修改后的数据记录。在原有数据表定义的基础上, 增加一个字段 HASHID, 例如对于表 1、表 2 中的数据表 T, 在历史库中创建一个对应历史数据表 T_his, 其在原始表 T 的基础上, 增加一个字段 HASHID, 与图 1 中的数据表数据记录版本管理表中的 HASHID 保持一致, 如表 3 所示。

表 3 数据表数据记录历史明细管理表示例

HASHID	A	B	C
000001	a	b	c
000002	a1	b	c

需特别指出的是, 当数据表元数据版本发生变化时, 需将现有历史库数据表 T_his 改名为 T_当前版本号_his, 并参照最新源系统表结构创建一个同名新表 T_his。

2.3 基于全量比对的数据起源标注生成和存储

参照第 2.1 和 2.2 节所描述的数据起源标注设计, 在数据中台项目实现了基于全量比对的数据起源标注生成和存储, 具体处理过程如图 2 所示。

为了生成必要的起源标注数据, 数据中台在针对每一个待抽取源系统数据表、执行每一次数据抽取任务时, 都必须将源系统数据表数据与原始库现有对应数据表数

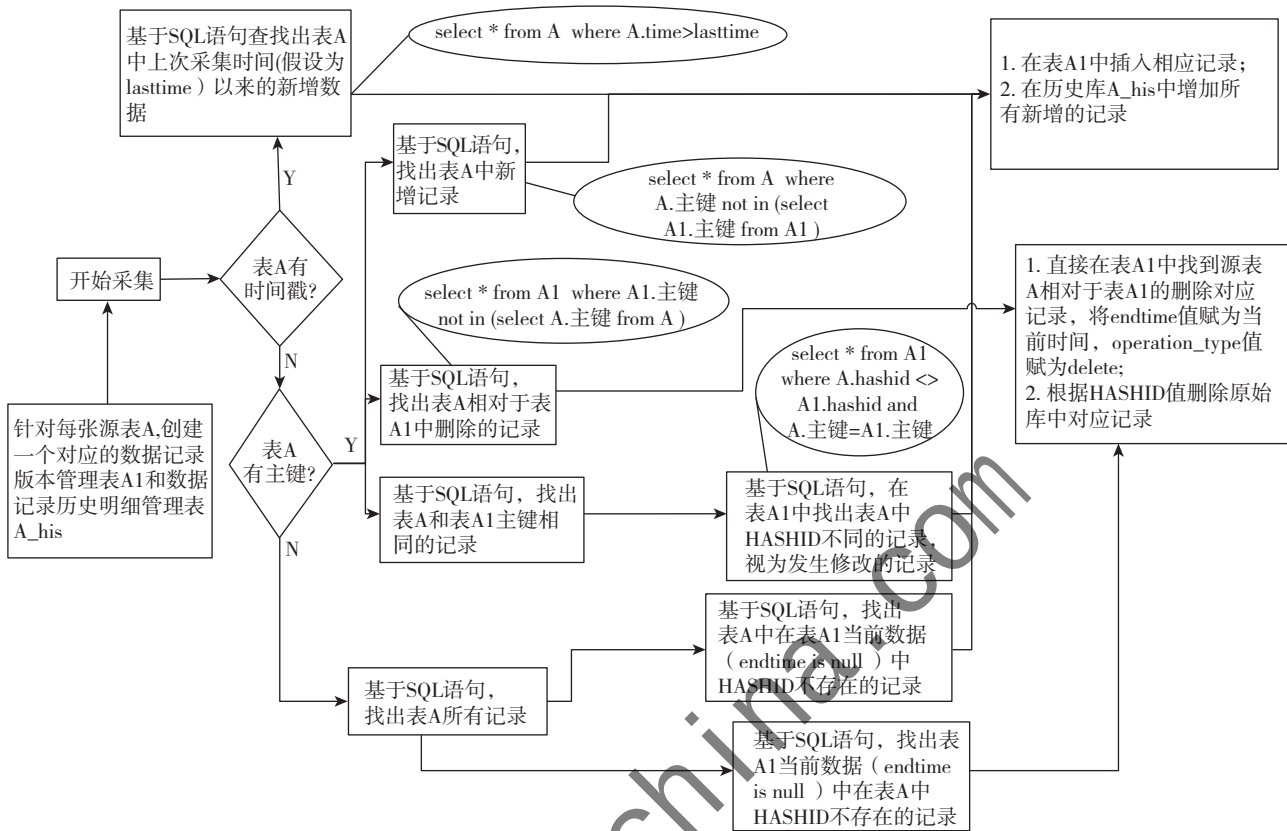


图2 基于时间戳和哈希对比的原始库数据表起源标注生成

据进行增量抽取。可分为三种情况进行比对：一是源系统数据表有时间戳，数据只增加不删除；二是源系统数据表无时间戳但有主键；三是源系统数据表既无时间戳也无主键。需要针对上述三种情况分别进行处理。

针对数据记录变化历史数据溯源需求，在数据中台项目专门建立了数据表数据记录版本管理库，为每一张原始库采集表建立了一张数据记录版本管理表（取名规则：采集表名_log），和原始库采集表一起存放于达梦数据库中；同时针对每一张原始库采集表建立了一张数据记录历史明细管理表（取名规则：采集表名_his），考虑到该表数据记录只增不删且累积数据量巨大，故将其存放于 NoSQL 数据库中（clickhouse/Hbase）。

3 数据表数据记录变化历史追溯

3.1 数据表元数据版本变化历史追溯

基于表1所示的数据表元数据版本管理表和表2所示的数据表元数据明细管理表，可以较好满足1.2节所描述的数据表元数据变化溯源的两类典型需求。

(1) 指定时间点 t_i 和数据表 T，按照条件 “begin_time < t_i and ((end_time > t_i and end_time < > null) or (end_time = null))”，对数据表元数据版本管理表进行过滤操作，

可获取时间点 t_i 的数据表 T 的版本号，然后基于数据表元数据明细管理表可获取对应数据表 “T_版本号” 的具体元数据；

(2) 指定时间段 (t_i, t_j) ，按照条件 “begin_time < t_i and ((end_time > t_j and end_time < > null) or (end_time = null))”，对数据表元数据版本管理表进行过滤操作，可获取时间段 (t_i, t_j) 的数据表 T 的所有版本号，然后基于数据表元数据明细管理表可获取对应所有版本的具体元数据。

3.2 数据表数据记录版本变化历史追溯

基于图1所示的数据表数据记录版本管理表和表3所示的数据表数据记录历史明细管理表，可以较好满足1.3节所描述的两类数据记录变化历史溯源需求。

(1) 指定数据表 T 中某条数据记录 m （例如图1中的 HASHID = “000002” 的记录），追溯得到指定时间点 t_i 的历史数据记录。首先参照3.1节给出的算法找出当时的元数据版本 “T_版本号”，假设为 T_1.0，然后基于数据表数据记录版本管理表（log 表），按照条件 “HASHID = m . HASHID” 进行过滤，获取符合条件的数据记录并获取其 “rele_HASHID” 值 A，然后以 “HASHID = A” 为

条件获取对应记录的 begintime 和 endtime, 递归执行直到满足 “begintime < t_i and ((endtime > t_i and endtime < > null) or endtime = null)”, 此时获取的数据记录即是数据表 T 中指定数据记录在时间点 t_i 的历史数据记录, 通过该条记录的 HASHID, 从对应的数据记录历史明细管理表中获取数据记录详细内容。

(2) 指定数据表 T 中某条数据记录 m , 追溯得到其在指定时间段 (t_i, t_j) 内的数据记录变化历史。首先按照上述方法找出 t_j 时间的历史数据记录, 然后继续递归执行, 直到满足 “begintime < t_i and ((endtime > t_j and endtime < > null) or (endtime = null)”, 此时获取的所有数据记录即是数据表 T 中指定数据记录 m 在时间段 (t_i, t_j) 的所有历史数据记录。每条历史数据记录均可查询获取对应的数据版本 (数据抽取时间)、变化类型 (新增/修改/删除) 和详细的数据内容。

(3) 指定数据表 T, 追溯得到其在指定时间点 t_i 的历史快照, 即该时间点所有数据记录。首先参照 3.1 节给出的算法找出时间点 t_i 的元数据版本 “T_版本号”, 假设为 T_1.0, 然后基于对应的数据表数据记录版本管理表 T_log, 按照条件 “begintime < t_i and ((endtime > t_i and endtime < > null) or (endtime = null)” 进行过滤, 即可获取时间点 t_i 数据表 T 的所有数据记录 HASHID, 关联对应的数据记录历史明细管理表, 即可获取数据表 T 在时间点 t_i 的历史快照。

3.3 指定时间段树形表演化历史追溯

在各种信息系统中, 描述编制体制的机构树是一种典型的常用树形基础数据表。本节以机构树形表为例, 来分析如何追溯指定时间段树形表数据演化历史。

机构 A 的任意下属机构的隶属关系变化都会导致机构 A 的树形表发生变化, 故机构 A 在指定时间段 [t_i, t_j] 内机构树的演化历史, 实质是机构 A 在时间点 t_i 的机构树快照加上时间段 [t_i, t_j] 内机构 A 的所有下属机构的隶属关系变化集合。

基于数据记录起源标注可以得到指定时间段 [t_i, t_j] 内针对机构 m 所有子机构的隶属关系变化, 这些变化的对象都是时间段 [t_i, t_j] 内机构 A 的子机构。追踪过程描述如下:

(1) 得到机构 A 在时间段 [t_i, t_j] 的直属子机构 m_k 的集合, 设为 $\Omega'(m, t_i, t_j)$, 其中 m 为 A 的根节点。按照 “begintime > t_i and ((endtime < t_j and endtime < > null) or (endtime = null)” 条件, 针对机构 A 对应的机构表数据记录版本管理表 (log 表) 进行过滤, 即可获取时间段 [t_i, t_j] 内机构 A 的数据记录变化。

如果 $\Omega(m, t_i, t_j) = \emptyset$, 可能是以下两种情况: 一

是 m 在时间段 [t_i, t_j] 内没有子机构, 二是 m 在时间段 [t_i, t_j] 内机构树结构未发生变化。

(2) 计算得到时间段 [t_i, t_j] 内子机构 m_k 隶属于 m 上的起始时间。假设 (‘配属’, t_{sn}) 表示子机构 m_k 在时间点 t_{sn} 时被归属于 m 下, (‘移除’, t'_{sn}) 表示子机构 m_k 在时间点 t'_{sn} 时被从 m 结构树中移除下来, $t_i \leq t_{sn}$, $t'_{sn} \leq t_j$ 表示 m_k 在时间段 [t_i, t_j] 一直是 m 子机构。共有四种情况, 如图 3 所示。

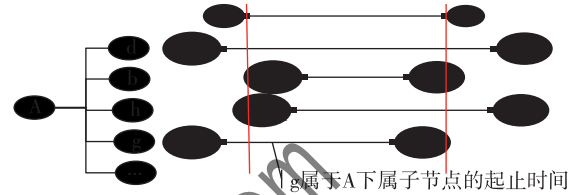


图3 机构A在 [t_i, t_j] 时间段内的4种子机构

①机构 m_d 在 [t_i, t_j] 时间段内一直是 m 的子机构, 可通过如下条件过滤得到:

$$\text{If } (t_{sn} \leq t_i \leq t_j \leq t'_{sn}) \vee (t_{sn} \leq t_i \leq t_j \wedge t'_{sn} \in \emptyset) \\ \text{then } t_{sd} = t_i, t_{ed} = t_j$$

假设 M_0 为此种机构的集合, 可通过递归运算得到 m_d 的机构树演化历史 $\Omega(m_d, t_{sd}, t_{ed})$ 。

②在时间段 [t_i, t_j] 内被归属于机构 m 又被移除下来的子机构 m_b , 可通过如下条件过滤得到:

$$\text{If } t_i \leq t_{sn} \leq t_j \leq t'_{sn} \\ \text{then } t_{sb} = t_{sn}, t_{eb} = t_j$$

假设 M_1 为这种机构的集合, 可通过递归运算得到 m_b 的机构树演化历史 $\Omega(m_b, t_{sb}, t_{eb})$ 。

③在时间段 [t_i, t_j] 内被归属到机构 m 上且一直未被移除下来的子机构 m_h , 可通过如下条件过滤得到:

$$\text{If } t_i \leq t_{sn} \leq t_j \leq t'_{sn} \\ \text{then } t_{sh} = t_{sn}, t_{eh} = t_j$$

假设 M_2 为这种机构的集合, 可通过递归运算得到 m_h 的机构树演化历史 $\Omega(m_h, t_{sh}, t_{eh})$ 。

④在时间段 [t_i, t_j] 内被从机构 m 移除下来的数据记录 m_g , 可通过如下条件过滤得到:

$$\text{If } t_{sn} \leq t_i \leq t'_{sn} < t_j \\ \text{then } t_{sg} = t_i, t_{eg} = t'_{sn}$$

假设 M_3 为这种机构的集合, 可通过递归运算得到 m_g 的机构树演化历史 $\Omega(m_g, t_{sg}, t_{eg})$ 。

(3) 得到机构 A 在 [t_i, t_j] 时间段内所有的机构树结构变化为: $M_0 + M_1 + M_2 + M_3$ 。

机构树、装备分类树和项目分类树等属于一种特殊的数据表, 一条数据记录代表一个机构 (装备分类、项目分类) 节点, 一般以层次码或者父子关系来标识节点

之间的父子关系。本节所描述的机构树演化历史算法也可用于装备分类树、项目分类树等树形结构表的历史数据溯源。

3.4 具体实现与应用

本方案在某大型单位数据中台系统中进行了具体实现。实验1: 指定时间段某数据表数据记录增删改总体情况, 以数据抽取任务时间为时间线, 统计每次抽取任务所对应的数据记录新增、修改、删除数量, 查看具体的数据记录。实验2: 指定时间段某数据表数据记录具体变化情况, 查看指定时间段当前数据表所有曾存储过的数据记录, 并查验每条数据记录对应的变化类型 (新增/修改/删除)。应用结果表明, 本文设计方案能够有效满足该单位数据中台原始库数据的数据溯源需求。由于需要对绝大部分源系统数据表和原始库数据表数据进行全量哈希比对, 在实际运行中面临数据量过大而导致的数据抽取速度偏慢问题, 通过采取负载均衡、并行处理等技术可以有效解决。

4 结论

对数据中台而言, 除原始库数据有数据溯源需求外, 实体库、主题库等数据也存在数据溯源需求。与原始库数据主要是对源信息系统数据进行同构化处理不同, 实体库、主题库等中台数据一般是基于两张或多张原始库数据表进行关联聚合而生成的, 这些数据的溯源需要考虑原始的关联聚合关系。可以有两种方案: 一是根据最新的源表数据生成实体表或主题表数据, 然后参照本文的方法生成对应的起源标注, 此方案同样需要生成大量的起源标注数据; 二是根据原始的关联聚合关系, 获得溯源时间点对应源表的历史快照, 然后基于源表历史快照生成实体表或主题表的历史快照, 此方法则无需生成大量的起源标注, 但对计算效率的要求相对较高, 未来可另文对此进行详细讨论和分析。

参考文献

[1] 任良全, 张力, 王建民. 产品维修阶段 BOM 数据起源研究 [J]. 计算机集成制造系统, 2010, 16 (10): 2187-2195.

- [2] Ren Genquan, Zhang Li, Wang Jianmin, et al. One method for provenance tracking of product lifecycle data in collaborative service environment [C]//2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC). IEEE, 2011: 347-356.
- [3] 杨莉. 面向关系数据库的数据起源研究与设计 [D]. 广州: 华南理工大学, 2017.
- [4] 樊治强. 基于可信时态关系的数据起源研究 [D]. 成都: 电子科技大学, 2022.
- [5] CHENEY J. Workshop on theory and practice of provenance event report [J]. ACM SIGMOD Record, 2009, 38 (2): 57-60.
- [6] 王芳, 赵洪. 数据溯源研究与实践进展 [J]. 情报学进展, 2020 (1): 313-353.
- [7] WPPDRIFF A, STONEBRAKER M. Supporting fine-grained data lineage in a database visualization environment [C]//Proceedings of 13th International Conference on Data Engineering, 1997. Birmingham: IEEE, 1997: 91-102.
- [8] BUNEMAN P, KHANNA S, TAN W C. Why and where: a characterization of data provenance [C]//8th International Conference on Database Theory (ICDT 2001). Berlin Heidelberg: Springer, 2001: 316-330.
- [9] 全国科学技术名词审定委员会. 计算机科学技术名词 (第三版) [M]. 北京: 科学出版社, 2018.
- [10] SIMMHAN Y L, PLALE B, GANNON D. Karma 2: provenance management for data-driven workflows [J]. International Journal of Web Services Research, 2008, 5 (2): 1-22.
- [11] 李亚子. 数据起源标注模式与描述模型 [J]. 现代图书情报技术, 2007 (7): 10-13.

(收稿日期: 2025-07-04)

作者简介:

周学文 (1978-), 男, 硕士, 高级工程师, 主要研究方向: 大数据分析。

薛猛 (1991-), 男, 硕士, 工程师, 主要研究方向: 数据工程。

版权声明

凡《网络安全与数据治理》录用的文章，如作者没有关于汇编权、翻译权、印刷权及电子版的复制权、信息网络传播权与发行权等版权的特殊声明，即视作该文章署名作者同意将该文章的汇编权、翻译权、印刷权及电子版的复制权、信息网络传播权与发行权授予本刊，本刊有权授权本刊合作数据库、合作媒体等合作伙伴使用。同时，本刊支付的稿酬已包含上述使用的费用，特此声明。

《网络安全与数据治理》编辑部

www.pcachina.com