使用iUAP DI进行基于数据源的变更数据捕捉（CDC）

基于源数据的CDC要求源数据库里面有相关的属性列，ETL的过程可以利用这些属性列，来判断出哪些数据时增量数据。最常见的属性列有两种:

**时间戳：**需要一个更新的时间戳，最好有两个时间戳，一个插入时间，一个最后一次更新时间。

**序列：**利用数据库的自增序列，很容易识别出新插入的数据。

这两种方法都需要一个额外的数据库表来存储上一次更新的时间或上次抽取的最后一个序列号。在实践中，一般是在一个独立的模式下创建这个参数表，不会在数据仓库和数据集市中创建。基于时间戳和自增序列的方法是CDC中最简单的实现方式，所以也是最常用的方法。但其缺点也是显而易见的，主要如下：

**区分插入和更新操作：**只有数据源中包含插入时间和更新时间两个时间戳时，才能区别是插入和更新操作，否则无法区分。

**删除记录的操作：**不能捕捉到物理删除操作，除非是逻辑删除，即记录没有真正的删除，只是立了个墓碑（做逻辑标志）；

**多次更新检测：**如果在一次同步周期内，数据被多次更新，只能同步最后一次更新的操作，中间的更新丢失了。

**实时能力：**时间戳和基于序列的数据抽取一般适用于批量操作，不适用于实时场景的数据加载。

在iUAP DI中使用时间戳的方式CDC例子：

首先，创建时间戳表CDC\_TIME,并出示化数据:

CREATE TABLE CDC\_TIME(

TABLENAME VARCHAR2(255) NOT NULL PRIMARY KEY

,LAST\_LOAD TIMESTAMP

,CURRENT\_LOAD TIMESTAMP

);

INSERT INTO CDC\_TIME VALUES('CUSTOMER',

to\_date('1970-01-01 00:00:01','yyyy-mm-dd hh24:mi:ss'),

to\_date('1970-01-01 00:00:01','yyyy-mm-dd hh24:mi:ss'));

其次，更新current\_load为当前时间。

再次，根据时间戳加载数据。

最后，修改current\_load和last\_load相同。

详细的作业截图如下：

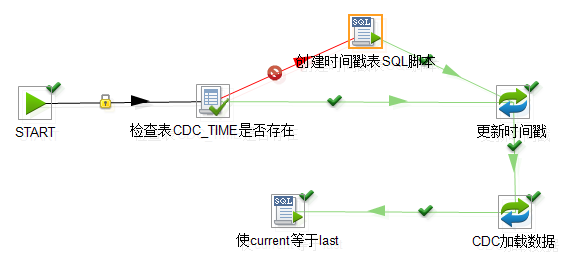


图1 作业流程图

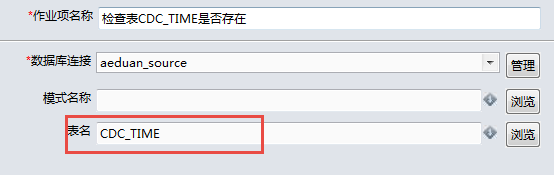


图2 检查表是否存在设置

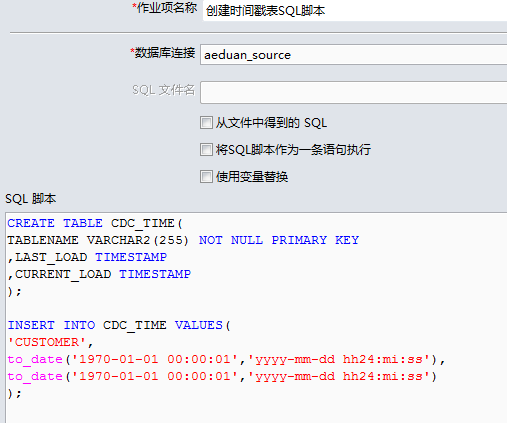


图3 创建时间戳表并初始化数据



图4 更新时间戳转换流程



图5 增量抽取数据转换流程



图6 增量抽取数据读取时间戳



图7 增量抽取数据读取数据



图8 增量抽取数据插入更新

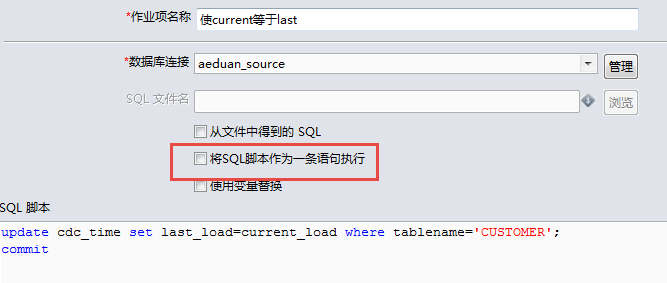


图9修改时间戳表current\_load和last\_load相同