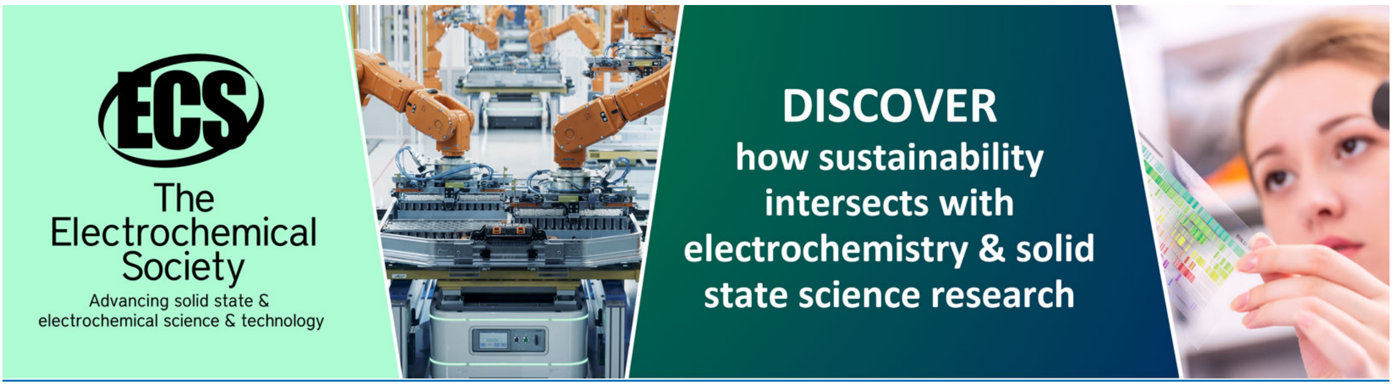
Design and Implementation of Oracle Database Incremental Data Capture Based on Trigger and Identification Table


ECS
The Electrochemical Society
Advancing solid state & electrochemical science & technology
DISCOVER
how sustainability intersects with electrochemistry & solid state science research

Design and Implementation of Oracle Database Incremental Data Capture Based on Trigger and Identification Table

Qiangqiang Hu*, Zhichun Gan and Bin Zhang

College of Information and Communication, National University of Defense Technology, Wuhan, Hubei, 430000, China

*Corresponding author's e-mail: 416389996@qq.com

Abstract. Data synchronization technology is one of the key technologies of data storage in distributed database. At present, it is an effective method to achieve data synchronization between source database and target database by capturing incremental data and transferring incremental files. Based on the research of Oracle database and the analysis of commonly using incremental data capture methods, this paper designed DML trigger and incremental data identification table to store incremental identification data in the identification table, and generated incremental XML files for network transmission according to the identification table. Finally, the feasibility of this method was tested. Tests showed that the method was reliable and stable, and could be applied to data synchronization in distributed databases.

1. Introduction

With the development of information technology, people's life has become colorful. At the same time, the requirement of database performance is increasing day by day. For enterprises, server downtime in a few minutes will cause significant losses for enterprises[1]. Data and its effective management are important business objectives of an organization[2].Distributed database improves the reliability and stability of data by multi-backup of database. At present, most platforms use distributed database. Database synchronization technology is a technology that keeps service data distributed in different locations consistent by replicating data. Due to the high requirement of network and machine performance for replicating full scale, the source database and target database are synchronized by transmitting incremental files at present. The incremental data synchronization process is shown in Figure 1.

Capturing incremental data and generating incremental files is a key link of database synchronization technology. At present, the common methods of capturing incremental data include trigger method, log analysis method, API middleware method, shadow table method, etc[3].

Oracle database is the most widely used database management system nowadays, which has great advantages for the storage of large data[4]. This paper studied Oracle data synchronization technology. Based on the research of incremental data capture technology and the characteristics of Oracle database, the incremental data capture method using trigger and identification table was determined. Combining with XML technology, an incremental XML file for network transmission was generated.

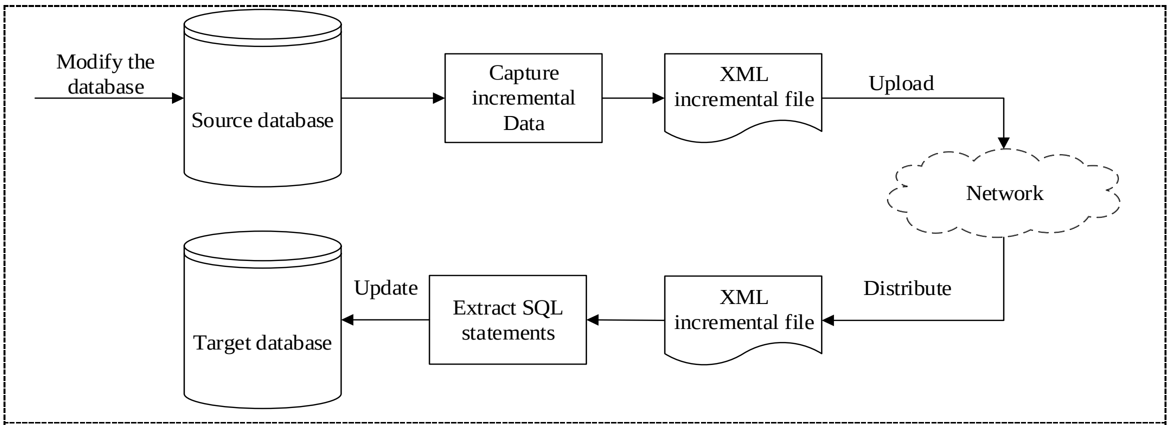


Figure 1. Schematic diagram of database incremental data synchronization process.

2. Incremental data capture method in database

Different methods will be used to capture incremental data according to different database architectures, data types and supporting standards. These methods are applied to different scenarios and can be used together.This section mainly explains the principle and implementation of shadow table method, and trigger method, and points out the applicable scenarios of each method.

2.1 Shadow table method

Shadow table method is also called table comparison method. By establishing a shadow table for each need change, incremental data are obtained by comparing data changes, and incremental changes are stored in the incremental scale[5]. Its implementation steps are shown in Figure 2.

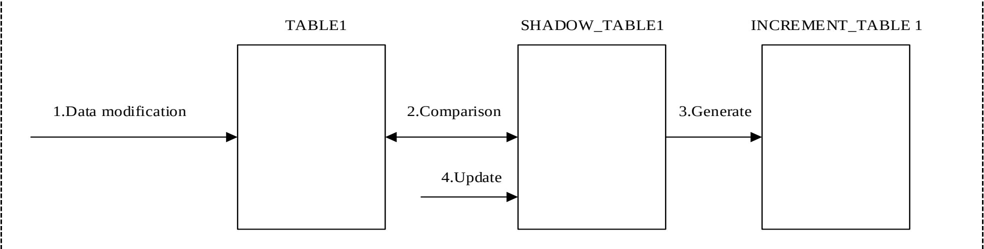


Figure 2. Diagram of shadow table method.

The key to this method is to find a way to compare the differences between the two tables. Oracle function minus and union can be used to compare the data.

For example, there were TABLE1 and its shadow table SHADOW_TABLE1. When TABLE1 table data changed, SHADOW_TABLE1 was the old table, TABLE1 was the new table. Using the SQL statement: select * from TABLE1 minus select * from SHADOW_TABLE1. You could get the data deleted by TABLE1; using the statement: select * from SHADOW_TABLE1 minus select * from TABLE1. You could get the data inserted and updated by TABLE1. The above results were combined by union function. The incremental data table could be obtained by inserting the results into the incremental scale and adding deletion or addition and update marks.

Shadow table method is the most intuitive way to obtain incremental data. However, there are some limitations on the timing of incremental data capture. And when the amount of data become large, the capture speed will be greatly reduced. This method is limited to databases with small amount of data. It can also be used to check the efficiency of other methods. It can also be used in combination with other methods to make incremental data more intuitive. For large databases, the establishment of shadow will increase the difficulty and load of database management. Trigger method has less impact on database than shadow table method.

2.2 Trigger Method

The trigger method is to create DML trigger into the data table[4]. When the data table was changed, the trigger taked effect and stored the incremental data in the incremental identification table. The identification table contains primary key information of incremental data and operation identification. Incremental data can be analyzed through joint identification tables and data tables. The identification table is divided into two categories: component identification table and unique identification table. Component identification means that each table that needs to capture changes generates an incremental identification table to form a full identification table. As shown in the Figure 3. The unique identifier table is to create only one identification table and store all changes in the identification table. As shown in Figure 4.

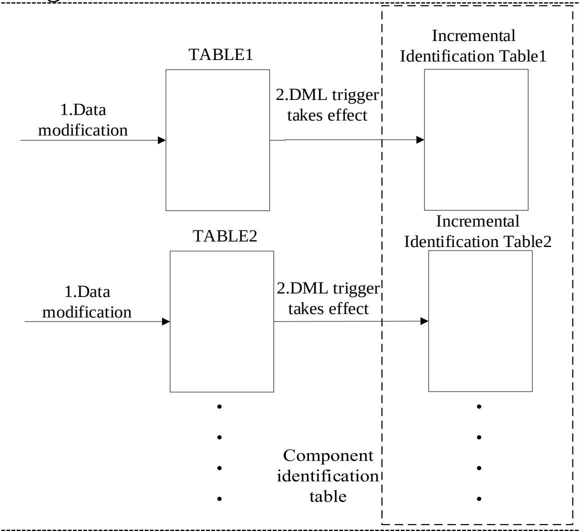


TABLE1
Incremental Identification Table1
1.Data modification
2.DML trigger takes effect
TABLE2
Incremental Identification Table2
1.Data modification
2.DML trigger takes effect
Component identification table

Figure 3. Diagram of trigger and component identification table method

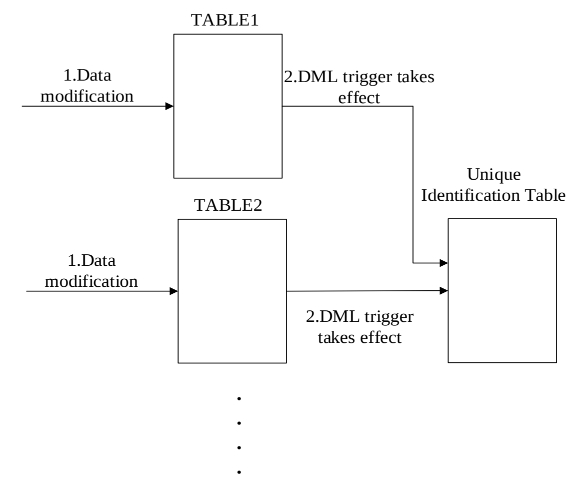


TABLE1
1.Data modification
2.DML trigger takes effect
Unique Identification Table
TABLE2
1.Data modification
2.DML trigger takes effect

Figure 4. Diagram of trigger and unique identification table method

The incremental identification of all tables can be stored by designing the identification table reasonably. The identification table generated by trigger method can have the same structure as that generated by shadow table method. The trigger method is more reasonable than the shadow table construction, and records the data operation of each item, so it has better expansibility.

Based on the above analysis, this paper designed an incremental data capture method based on trigger and unique identifier table.

3. Design and Implementation

3.1 Design of Identification Table

The method of using triggers requires that the captured incremental data was stored in the identification table. Establish an identification table FLAG_TABLE in the database. The table stored the data table names, the primary keys and the primary key values, as well as the DML operation types and trigger times. The primary key was set to a sequence, and the value of each record was increased by 1. The design of FLAG_TABLE table is shown in the Table 1.

Table 1. Design of identification table FLAG_TABLE.

COLUMN_NAME	DATA_TYPE	NULLABLE	Primary key	Notes
RECORD	INT	No	Yes	Record number
T_NAME	VARCHAR	Yes	No	Source table name
T_KEY	VARCHAR	Yes	No	Source table primary key name
T_KEYVALUE1	VARCHAR	Yes	No	primary key value

				ofNew table
T_KEYVALUE2	VARCHAR	Yes	No	Primary key value of original table
D_TYPE	VARCHAR	Yes	No	DML operation type
D_TIME	VARCHAR	Yes	No	DML Operation time

3.2 Design of Trigger

DML trigger includes insert trigger, delete trigger and update trigger. Combining with TEST_TABLE structure of experiment table, insert trigger, delete trigger and update trigger were edited and inserted respectively. As shown in Table 2.

Table 2. TEST_TABLE.

COLUMN_NAME	DATA_TYPE	NULLABLE	Primary key	Notes
NUM	VARCHAR	No	Yes	Serial number
USERNAME	VARCHAR	Yes	No	User name
PASSWORD	VARCHAR	Yes	No	Password
DEPARTMENT	VARCHAR	Yes	No	department

3.2.1 Insert triggers

The trigger is set to trigger after the insert operation, and the trigger is recorded for each statement. The results are stored in the designed identification table. The code to create the insert trigger is as follow:

```
CREAT  OR REPLACE  TRIGGER TRIGGER_I
AFTER INSERT ON TEST_TABLE
FOR EACH ROW  BEGIN
INSERT INTO FLG_TABLE(T_NAME, T_KEY, T_KEYVALUE, D_TYPE,D_TIME)
VALUES('TABLE', 'NUM', :NEW.NUM, 'I', systimestamp);
END;
```

3.2.2 Delete trigger

The code to create the delete trigger is as follow:

```
CREAT OR REPLACE TRIGGER TRIGGER_D
AFTER DELETE ON TEST_TABLE
FOR EACH ROW BEGIN
INSERT INTO FLG_TABLE(T_NAME, T_KEY, T_KEYVALUE2, D_TYPE, D_TIME)
VALUES('TABLE', 'NUM', :OLD.NUM, 'D', systimestamp);
END;
```

3.2.3 Update trigger

The code to create the update trigger is as follow:

```
CREAT OR REPLACE TRIGGER TRIGGER_U
BEFORE UPDATE OF NUM, PASWER, USERNAME ON TEST_TABLE
FOR EACH ROW BEGIN
INSERT INTO
FLG_TABLE(T_NAME, T_KEY, T_KEYVALUE1, T_KEYVALUE2, D_TYPE, D_TIME)
VALUES('TABLE', 'NUM', :NEW.NUM, :OLD.NUM, 'U', systimestamp);
END;
```

3.3 Test verification

Fill in the experimental data in TEST_TABLE . As shown in Table 3.

Table 3. TEST_TABLE.

NUM	USERNAME	PASSWORD	DEPARTMENT
1	qqh	qqh	network planning
2	wy	wy	network control
3	swj	swj	network monitoring

Perform DML operations:

- Insert data:4, zyf, zyf, network planning
- Delete data:2, wy, wy, network control
- Update data:1, qqh, qqh, network planning. Change the password value to 123456.

The updated data of TEST_TABLE is shown in Table 4.

Table 4. Updated TEST_TABLE.

NUM	USERNAME	PASSWORD	DEPARTMENT
1	qqh	123456	network planning
3	swj	swj	network monitoring
4	zyf	zyf	network planning

When the above operation is performed, the trigger records the change data and fills in the identification form FLAG_TABLE. The data in FLAG_TABLE are shown in Table 5.

Table 5. FLAG_TABLE.

RECORD	T_NAME	T_KEY	T_KEYVALUE1	T_KEYVALUE2	D_TYPE	D_TIME
1	TEST_TABLE	NUM	4	Null	I	21-3 -19 04.07
2	TEST_TABLE	NUM	Null	2	D	21-3 -19 04.07
3	TEST_TABLE	NUM	1	1	U	21-3 -19 04.08

The data in the identification table FLAG_TABLE reflects the change of the source table. By combining the identification table FLAG_TABLE with Updated TEST_TABLE, incremental data can be obtained.

4. XML Incremental file generation

Incremental data can be extracted from source data and identification tables and stored in XML files. XML is a language with uniform format and supports cross-platform operation. XML can be easily stored and transmitted in the network[6]. The value of field d_type is obtained by querying the record of the identification table FLAG_TABLE. If the value is D, you only need to get the value of T_KEYVALUE2 in the identification table, and if it is I or U, you need to associate the data of the source table. According to the XML format, it is stored in the XML file one by one, that is to say, incremental XML file generation is completed. The flow chart is shown in Figure 5.

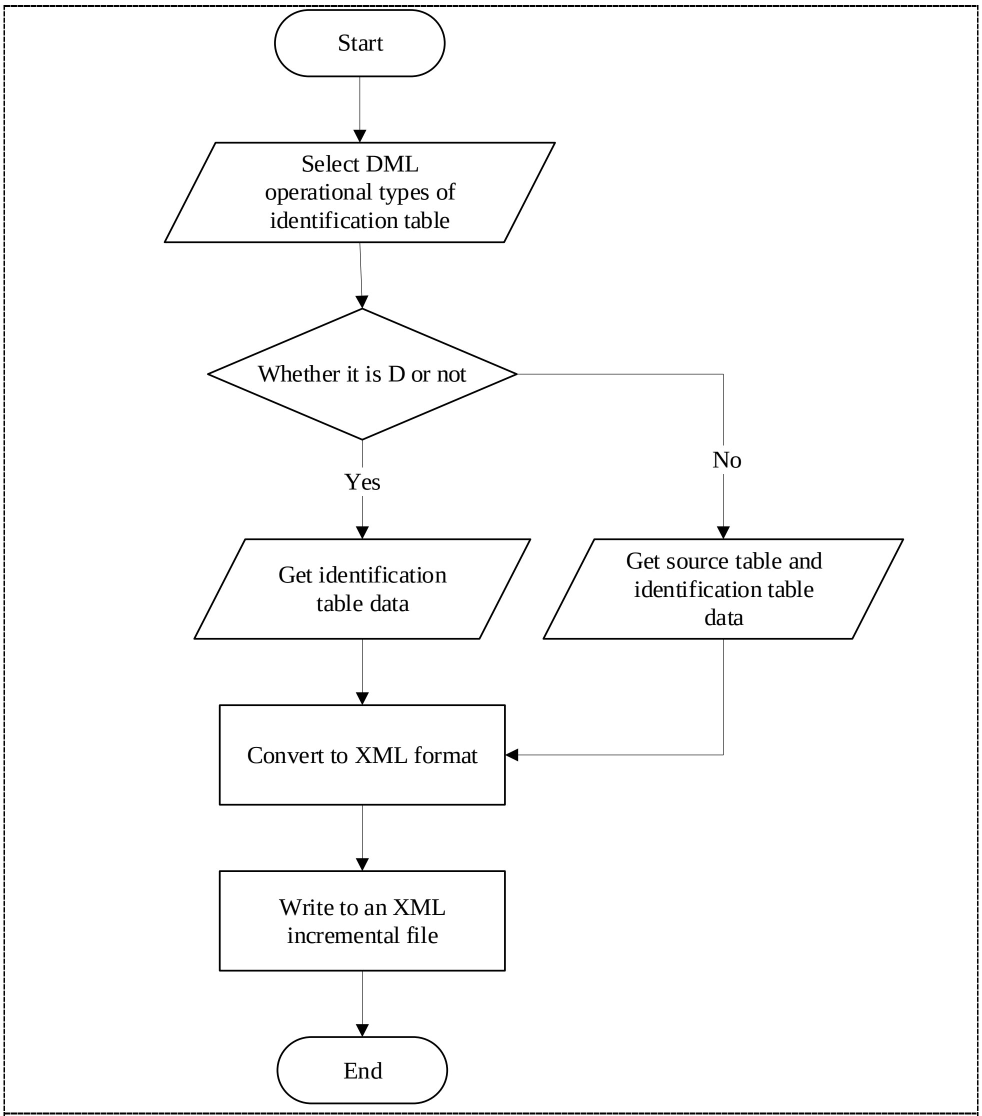

Start
Select DML operational types of identification table
Whether it is D or not
Yes
No
Get identification table data
Get source table and identification table data
Convert to XML format
Write to an XML incremental file
End

Figure 5. XML file generation flow chart.

Take the source table data TABLE 4 and the identification table TABLE 5 as examples. The format of the XML file generated by the source table execution insertion statement is as follow:

```
<?xml Version=”1.0” encoding=”utf-8”>
<Incremental_DataFile001>
      <Data  RECORD=1>
            <T_NAME>TEST_TABLE</T_NAME>
            <T_KEY>NUM</T_KEY>
            <T_KEYVALUE1 NUM=”4”>
                        <USERNAME> zyf</USERNAME>
                        <PASSWORD> zyf</PASSWORD>
                        <DEPARTMENT> network planning</DEPARTMENT>
            </T_KEYVALUE1>
            <D_TYPE>I</D_TYPE>
            <D_TIME>21-3 -19 04.07</D_TIME>
      </Data>
</Incremental_DataFile001>
```

5. Summary

In this paper, the principle of Oracle database incremental data capture was studied, and the method of trigger and identification table was proposed. By designing the incremental identification table and editing trigger, the source data table was updated and the incremental data was stored at the same time. Then the incremental data was stored in the XML file by combining the structured representation of XML. XML files could be parsed and restored to SQL statements, which could act on the target database to achieve synchronous updates of the database. Through programming experiments, it could be concluded that the incremental data acquisition method based on trigger and identification tables was stable and efficient, and could be well applied in distributed database. However, citation triggers and identification tables have a certain impact on the database, which will increase the load of the database. For database system with too much data, we can not rely too much on this method. It should also be used in conjunction with other echnologies.

References

[1] Liu J.(2013), Oracle Database Disaster Recovery System for Securities Industry. Fudan University, Shanghai.
[2] Singh S.K.(2010) Database System Concept, design and Application . Machinery Industry Press, Bei jing.
[3] Cui H. X. ,Feng J.,Ma W.J.(2016) Research and Implementation of Heterogeneous Database Synchronization Technology Based on Isolated Gate. Software Engineering,19(2):10-13
[4] Cheng C. B.,Zhang Y.B.(2016) Oracle 12C Chinese Version Database Management Application and Development Practice Course. Tsinghua University Press, Beijing.
[5] He W., Hao Y.Q., Zhou L.B.(2010) Database Synchronization Method Based On Network Isolation. Information Security and Communication Secrecy, 1: 82-84.
[6] Zhang Z.(2010) Lightweight Differential Incremental Backup System Based on File Matching.Shanghai Jiaotong University,Shanghai.