

Removal of History table from Database & Update Command from SQL & Edit option from GUI

Shubhankar Paul

Abstract— “Update” is a command in database language like SQL etc. to update a/some column value of a particular row of a given table leaving all the other column unchanged in that particular row of that given table. We will discuss in this paper how to remove “Update” command from the database language like SQL and also “History” table from the database by adding another column removing also a column “LAST_UPDATED” from a given table. We will also discuss the process from GUI level.

Index Terms— SQL, GUI level, database language.

I. INTRODUCTION

A. Content

- 1) What is “Update” command.
- 2) Example.
- 3) What is “History” table.
- 4) Example.
- 5) How to remove “Update” command and “History” table.
- 6) Removal of EDIT/UPDATE option from GUI.
- 7) Advantages.
- 8) Pareto Efficiency.
- 9) Pareto improvement in existing system.
- 10) Invention is efficient or not.

II. WHAT IS “UPDATE” COMMAND

In SQL “Update” command is used to update a/some column value of a particular row of a given table by leaving other column values of that particular row of the given table unchanged.

Example

Let us consider the following data of a company in database.

EMPLOYEE_TABLE :

ROW_ID	EMP_ID	NAME	JOB_ROLE
1	1122	John	Officer
2	5972	Jack	Sr.Officer
3	9072	Paul	Jr.Officer

We have considered 3 rows and 4 columns. Now, let say, JOB_ROLE of Paul is promoted to “Officer” from “Jr. Officer”. Here we will use the below command to update the JOB_ROLE only leaving ROW_ID, EMP_ID, NAME unchanged.

```
UPDATE EMPLOYEE_TABLE
SET JOB_ROLE = 'Officer'
WHERE EMP_ID = '9072';
```

And the result will be as given below in the image :

EMPLOYEE_TABLE :

ROW_ID	EMP_ID	NAME	JOB_ROLE
1	1122	John	Officer
2	5972	Jack	Sr.Officer
3	9072	Paul	Officer

This is how “Update” command works.

III. WHAT IS “HISTORY” TABLE

“History” table stores all previous data of a row which has been updated one or multiple time(s).

Example

Let us consider the following data of employees of a given company :

EMPLOYEE_TABLE :

ROW_ID	EMP_ID	NAME	J_ROLE	LAST_UPD
1	1122	John	Officer	11/5/2003
2	5972	Jack	Sr.Officer	9/8/2004
3	9072	Paul	Jr.Officer	5/8/2005

Manuscript received December 02, 2013.

Shubhankar Paul, Passed BE in Electrical Engineering from Jadavpur University in 2007. Worked at IBM as Manual Tester with designation Application Consultant for 3 years 4 months.

Let, Paul is promoted to "Officer" from "Jr. Officer" on 3/4/2006 and Jack is demoted to "Officer" from "Sr. Officer" on 1/2/2005. The update command will change the EMPLOYEE_TABLE as below :

EMPLOYEE_TABLE :

ROW_ID	EMP_ID	NAME	J_ROLE	LAST_UPD
1	1122	John	Officer	11/5/2003
2	5972	Jack	Officer	1/2/2005
3	9072	Paul	Officer	3/4/2006

Now, the "History" table will store the previous records which were previously and updated as shown below :

HISTORY_EMP_TABLE :

ROW_ID	EMP_ID	NAME	J_ROLE	LAST_UPD
1	5972	Jack	Sr.Officer	9/8/2004
2	9072	Paul	Jr.Officer	5/8/2005

How to remove "Update" command and "History" table: We can do it in two ways. We will define a new operator, say, "Cross" or we can add a new column to the existing table and remove LAST_UPD.

1. Cross operator :

a(cross)b means the record whose ROW_ID (primary key) = a has been updated (b-1) times and 1 time it was created. For example, 2(cross)5 means second record is created and updated 4 times after creation. 5(cross)2 means fifth record created and updated 1 time after creation. So, 2(cross)5 is not equal to 5(cross)2 (not to confuse with general multiplication where $2*5 = 5*2 = 10$).

2. Adding one column and removing LAST_UPD column :

The column that will be added will contain value 1 when a record is created and will take values 2, 3, 4, ... when updated.

So, Primary key of a record in a given table will be composed of 2 columns viz. ROW_ID and NEW_COLUMN. When a record will be searched it will show the last numbered NEW_COLUMN. We have seen in the Example of "History" table we need LAST_UPD column to track when it was last updated. Here we will see in an example with previous data how NEW_COLUMN will help unnecessary of "History" table and "LAST_UPD" column.

EMPLOYEE_TABLE :

ROW_ID	NEW_COLUMN	NAME	J_ROLE	UPD_DATE
1	1	John	Officer	11/5/2003
2	1	Jack	Sr.Officer	9/8/2004
3	1	Paul	Jr.Officer	5/8/2005

Please note UPD_DATE is not a new column. It should exist in every table to track when the record was created. Now, Paul is promoted to "Officer" from "Jr. Officer" on 3/4/2006 and Jack is demoted to "Officer" from "Sr. Officer" on 1/2/2005. Without using "Update" command we will use "INSERT INTO" as below :

```
INSERT INTO EMPLOYEE_TABLE (ROW_ID,
NEW_COLUMN, NAME, J_ROLE, UPD_DATE)
SELECT ROW_ID, NEW_COLUMN, NAME
FROM EMPLOYEE_TABLE
WHERE ROW_ID = 2
SET J_ROLE = 'Officer', NEW_COLUMN =
'NEW_COLUMN+1';
```

This command will insert a new row in the table with same ROW_ID and all other column except J_ROLE which is updated to 'Officer' and NEW_COLUMN is set to 2.

Similarly, for Paul,

```
INSERT INTO EMPLOYEE_TABLE (ROW_ID,
NEW_COLUMN, NAME, J_ROLE, UPD_DATE)
SELECT ROW_ID, NEW_COLUMN, NAME
FROM EMPLOYEE_TABLE
WHERE ROW_ID = 3
SET J_ROLE = 'Officer', NEW_COLUMN =
'NEW_COLUMN+1';
```

The updated table will look like below :

EMPLOYEE_TABLE :

ROW_ID	NEW_COLUMN	NAME	J_ROLE	UPD_DATE
1	1	John	Officer	11/5/2003
2	1	Jack	Sr.Office	9/8/2004
2	2	Jack	Officer	1/2/2005
3	1	Paul	Jr.Officer	5/8/2005
3	2	Paul	Officer	3/4/2006

We see that History is already created in the UPD_DATE or CREATE_DATE. All records are there in this table. So, we do not need any "History" table.

IV. REMOVAL OF EDIT/UPDATE OPTION FROM GUI

In GUI we will use "Copy Record" option to update a record. When we will click on "Copy Record" option then that particular row will be copied with NEW_COLUMN = NEW_COLUMN+1 value and it will be editable until saved. Then we can put updated value for a/some column we want to update. Then it will be saved in database with same ROW_ID but different NEW_COLUMN. As we have already discussed primary key will be composition of ROW_ID and NEW_COLUMN.

Note : We can also define "Copy Row" in database level also like GUI.

V. ADVANTAGES

1. A table space (History table) is removed without affecting anything else. This will enhance the efficiency of database.
2. A command "Update" is removed from database language like SQL.
3. EDIT/UPDATE option is removed from GUI. Sometimes it happens due to unconsciousness a data is updated. This type of error is removed as it will be read-only in GUI as well for ADMIN.
4. Removal of an option from GUI (EDIT/UPDATE) will increase the efficiency of Business layer.
5. Every record that is deleted will be given a NULL or 0 value in NEW_COLUMN. It will help the system to search for the maximum value of NEW_COLUMN otherwise the system will not be able to understand the maximum value of NEW_COLUMN. For example, say a record is created and updated 2 times and second record i.e. the record with NEW_COLUMN = 2 value is deleted. Then query

command will not find 3 as after 1 it will not find 2. But if it finds a NULL or 0 for NEW_COLUMN then it will understand record 2 is deleted. Then it will search for record 3 and output the maximum valued NEW_COLUMN record. We can calculate the number of record with NEW_COLUMN = NULL or 0. Now $\{(Total\ number\ of\ record - Record\ with\ NEW_COLUMN = NULL\ or\ 0) / Total\ number\ of\ records\} * 100$ will give the consciousness of a team working on a particular project. Because we are not supposed to delete record if it is not mistyped any column due to unconsciousness.

6. It is convenient to use 2 digit integer for NEW_COLUMN because a record will not be updated for more than 99 times. It will reduce the database level memory as the column LAST_UPDATED, which is a date-time stamp, is more memory occupying, is removed.
7. The more NEW_COLUMN value the interaction of that customer is more with the company as the data is getting updated for NEW_COLUMN - 1 times. NEW_COLUMN value can also be greater if record is deleted due to unconsciousness of an employee as discussed in point 5. So, we can define a measure to calculate interaction of a customer as, $Interaction = \{(Maximum\ NEW_COLUMN\ value - number\ of\ records\ with\ NEW_COLUMN = NULL\ or\ 0\ with\ given\ ROW_ID) / Maximum\ NEW_COLUMN\ value\} * 100\%$. The more the value the interaction of that particular customer with company is more.
8. The efficiency of GUI also increase as an option is removed from there.
9. If we define cross operator then NEW_COLUMN is not needed.

VI. PARETO EFFICIENCY

We start with the following definition: if we can find a way to make some people better off without making anybody else worse off, we have a Pareto improvement. If an allocation allows for a Pareto improvement, it is called Pareto inefficient; if an allocation is such that no Pareto improvements are possible, it is called Pareto efficient.

In our given system we are not making the system worse off in any condition and we can find lots of advantages as discussed. So, there is a Pareto Improvement in the system is possible. According to above definition the system is Pareto inefficient.

Invention is Efficient or not :

From the Pareto Efficiency section we can see that we find a Pareto improvement. So, the invention is efficient.

REFERENCES

- [1] Intermediate Microeconomics A modern approach 8th edition by Hal. R. Varian.
- [2]<http://stackoverflow.com/questions/9156340/how-to-copy-a-row-and-insert-in-same-table-with-a-autoincrement-field-in-mysql>

Shubhankar Paul, Passed BE in Electrical Engineering from Jadavpur University in 2007. Worked at IBM as Manual Tester with designation Application Consultant for 3 years 4 months. Worked at IIT Bombay for 3 months as JRF. Published 2 papers at International Journal.