

Re: newbie SP question

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.mseq/2004-12/0020.html>

From: Hugo Kornelis (*hugo_at_pe_NO_rFact.in_SPAM_fo*)

Date: 12/20/04

Date: Mon, 20 Dec 2004 22:27:49 +0100

On Mon, 20 Dec 2004 06:13:04 -0800, Miguel Salles wrote:

>Thats what I want to do: I have a staging table were I load all my products
>(dbo.product_laam) I want to read the data from it and populate a new table
>that has Product_ID as PK (wich Identify a single product).
>
>--I want to insert new data in my product table as:

(snip)

>--Then I want to update existing data in my table as:

(snip)

>--All in one Stored Procedure

Hi Miguel,

Well, you're almost there. Since a stored procedure can contain many statements, just popping these two statements in one stored proc would give you a good start.

More advanced would be to reverse the order (first update, then insert) to speed up execution (if you insert first, the newly inserted rows get updated as well). Also, you might want to enclose both statements in a transaction so that all modifications can be rolled back if anything goes wrong.

I also used CURRENT_TIMESTAMP (ANSI-standard SQL syntax) instead of your {fn NOW()} (not part of SQL at all, though it does seem to work - I believe that this is caught by the ODBC driver, but I'm not sure and I couldn't find it in BOL), and I used a slightly clearer (IMO) version of the UPDATE FROM syntax. You do know that this syntax is not portable, do you?

I also added the column list to the INSERT command. Using INSERT without columns list is, just as using SELECT *, a sure way to break your

application when you least expect it. My final modification was to replace the NOT IN with an EXISTS subquery. Both should work equally well; the reason I always prefer NOT EXISTS is that you don't get the results you want if NULLS can be present in the subquery, so I just totally shun NOT IN. If performance matters, you should test if using an outer join instead of NOT EXISTS is quicker.

```
CREATE PROC Salles
AS
BEGIN TRANSACTION
UPDATE T
SET toy_number = T1.toy_number
    , master_item_description = T1.master_item_description
    , product_category_1 = T1.product_category_1
    , product_category_2 = T1.product_category_2
    , product_category_3 = T1.product_category_3
    , product_category_4 = T1.product_category_4
    , product_category_5 = T1.product_category_5
    , category_1_description = T1.category_1_description
    , category_2_description = T1.category_2_description
    , category_3_description = T1.category_3_description
    , category_4_description = T1.category_4_description
    , category_5_description = T1.category_5_description
    , last_update_date = CURRENT_TIMESTAMP
FROM DATAWAREHOUSE.rsc_dim_product_agg_laam AS T
INNER JOIN dbo.product_laam AS T1
    ON T.product_id = T1.product_id
IF @@ERROR <> 0
    GOTO Wrong

INSERT INTO DATAWAREHOUSE.rsc_dim_product_agg_laam
    ( product_id
    , toy_number
    , master_item_description
    , product_category_1
    , product_category_2
    , product_category_3
    , product_category_4
    , product_category_5
    , category_1_description
    , category_2_description
    , category_3_description
    , category_4_description
    , category_5_description
    , creation_date --- ???
    , last_update_date --- ???
select T1.product_id
    , T1.toy_number
    , T1.master_item_description
    , T1.product_category_1
    , T1.product_category_2
```

```
, T1.product_category_3
, T1.product_category_4
, T1.product_category_5
, T1.category_1_description
, T1.category_2_description
, T1.category_3_description
, T1.category_4_description
, T1.category_5_description
, CURRENT_TIMESTAMP
, NULL
FROM dbo.product_laam AS T1
WHERE NOT EXISTS
(SELECT *
 FROM DATAWAREHOUSE.rsc_dim_product_agg_laam AS T
 WHERE T.product_id = T1.product_id)
IF @@ERROR <> 0
  GOTO Wrong
```

Rite:
COMMIT TRANSACTION
RETURN 0

Wrong:
ROLLBACK TRANSACTION
RAISERROR ('Somethign is rotten in the state of Danmark', 16, 1)
RETURN -1
go

(Note: this is untested!!)

(Second note: with column names including category_1 through 5 and their descriptions, you might want to look into normalizing your design. How sure can you be that there will never be a sixth category??)

Best, Hugo

--
(Remove _NO_ and _SPAM_ to get my e-mail address)