Magic xpi 4.13

データマッパー UPSERTの使い方

マジックソフトウェア・ジャパン株式会社





目次:データーマッパー UPSERTの使い方

・<u>第1章 UPSERTの概要</u>

1.1 UPSETとは 1.2 サポートするデータベース 1.3 UPSERTの仕組み

第2章 UPSERTの使い方

2.1 データマッパー:送り先 DataBaseのプロパティ UPSERT
2.2 データベースウィザード
2.3 データベースウィザードで生成されたSQL文の修正





UPSERTの概要



All Rights Reserved, Copyright(C) Magic Software Japan K.K. 2020

1.1 UPSERTとは





1.1 UPSERTとは

Magic xpi のデータマッパーで送り先にDatabaseを配置し、
 そのプロパティのUPSERTをYesに設定すると、テーブルに対してUPSERT処理
 を行う事ができます。

UPSERTとは、「データがあればUPDATE、なければINSERTする」という処理です。 各DBMSにより、UPSERT用のSQLが用意されており、そのSQLを Magic xpiのデータマッパーで 発行することで、UPSERT処理が実現されます。



1.1 UPSERTとは

・ UPSERTとは、データの新規挿入(INSERT)ができれば挿入を行い、 新規挿入ができなければ更新(UPDATE)を行います。



naai

1.2 サポートする データベース





1.2 サポートするデータベース

DatabaseTriggerがサポートするデータベースは下記の通りです。

DBMS	バージョン
MS-SQL	2008, 2008R2, 2012, 2014, 2016, 2017,2019
Oracle	12c, 18c, 19c
DB2/400	V7R1, V7R2, V7R3, V7R4



1.3 UPSERTの仕組み





1.3 UPSERTの仕組み

- ・ 各DBMSで用意されている、UPSERT用のSQL文を組み立て、マップで線を引くことで、 UPSERT用のSQLが発行されるように組み込みます。
- ・ Microsoft SQLServer 、 Oracle 、 DB2/400の場合、MERGE文を使います。
- ・ MERGE文の基本構文:

MERGE INTO 主表 USING 副表 ON (条件)
WHEN MATCHED THEN
UPDATE SET 列1 = 値1 [, 列2 = 値2 ...]
WHEN NOT MATCHED THEN
INSERT (列1 [, 列2 ...]) VALUES (値1 [, 値2 ...])





UPSERTの使い方



All Rights Reserved, Copyright(C) Magic Software Japan K.K. 2020

2.1 データマッパー 送り先 DataBaseの プロパティUPSERT



All Rights Reserved, Copyright(C) Magic Software Japan K.K. 2020

2.1 データマッパー:送り先 DataBaseのプロパティ UPSERT

- データマッパーの送り先にDatabaseを配置し、そのプロパティのUPSERTをYesに変更します。
- データベースウィザード機能を利用して、SQL文を生成します。

(DBMSに則したUPSERT用SQL文の生成を支援してくれます)

DataBase_2 - SQL文が望自です。テータハーススキーマ アーカル型 アーク クリック マイナーマ情報 クリック D 2 マー般 マイサード ウイサード マー般 ウトス定義 MicrosoftSQLServer_DB SQL文 DataBase_2 説明 アークバースウィザード 直面に遷移	R DBオペレーションを選択してください: UPSERT ジ ジ 除件 DBオペレーションを選択してください: UPSERT ジ ジ ジ ジ ジ ジ ジ ン ジ ン ジ ン ジ ン ジ ン ジ ン ジ



キャンセル

戻る

次へ

終了



All Rights Reserved, Copyright(C) Magic Software Japan K.K. 2020

2.2 データベース ウィザード

 データベース ウィザード画面に従って進めると、各DBMSに則したUPSERT用のSQL 文が生成されます。

次ページから、データベースウィザードの操作画面を説明します。



2.2 データベースウィザード(1)

・データベースがMicrosoft SQLServerの場合の操作画面例:

データベースウイザード	>	× 🕒 データベースウィザード				
	・ :選択できます。 のみ選択できます。 ・	★	DBオペレーションを選択してください: UPSERT 送り元オブジェクトでは、複数のテーガルを選択できます。 送り元オブジェクトでは、複数のテーブルを選択できます。 、送りたオブジェクトの場合、テーブルは1つのみ選択できま つフルク定義のみ	ŧŧ.		
¹⁾ 有効なテーガル [[FESTTable1] (TESTTable2] [TESTTBL1]	選択演テープル	次へ 979	有効なテーブル (TESTTable2) (TESTTBL1)	選択) dbo (追加)> (利除)	湾テーブル (TESTTable1)	
	戻る 次へ 終了 キャンセル					

中央左の「有効なテーブル」のリストボックス から、目的のテーブルをクリックして選択し、 [追加]をクリックします。 [次へ]をクリックす る。 選択が完了したら、[次へ]をクリックします。

2.2 データベースウィザード(2)

・データベースがMicrosoft SQLServerの場合の操作画面例:



使用可能列のリストボックスのテーブル横の▼ をクリックして展開し、必要なカラムをクリッ クして選択し、[追加]ボタンをクリックします。



All Rights Reserved	Copyright(C)	Magic Software	Japan K.K. 2020
---------------------	--------------	----------------	-----------------

OUTPERFORM THE FUTURE

テーブル選択 列選択	選択したテーブルのマッピングは、以下で選択した列を フィールド表示項: Database Order v	使用して作成されます。		
UPSERTの条件 Join 文 サマリ	使用可能列	追加 > < 約除	遵択资列 ▼ dbc[TESTTable1] [D] [String1] [Date1] [Time1] [Log1]	

列(カラム)の選択が完了したら、[次へ]を クリックします。

2.2 データベースウィザード(3)

・データベースがMicrosoft SQLServerの場合の操作画面例:

ウィザード マッピング定員に対するUPSERT条件を設定します。 使用可能列 doo.[TESTTable1][J0] doo.[TESTTable1][J0[1] doo.[TESTTable1][J0[1] doo.[TESTTable1][J0[1] UPSERT/0条件 doo.[TESTTable1].J0D] = < 71D7 >	×	
マルビッグ定義に対するUPSERT条件を設定します。 使用可能列 案 dob.0[TESTTable1][0] dob.0[TESTTable1][String1] dob.0[TESTTable1][Date1] dob.0[TESTTable1][Ime1] dob.0[TESTTable1][Ime1] dob.0[TESTTable1][Ime1] dob.0[TESTTable1][Ime1] dob.0[TESTTable1][Ime1]		● データベースツザード
dbo.(TESTTable1),500) dbo.(TESTTable1),500;g1) dbo.(TESTTable1),000;g1) dbo.(TESTTable1),100;g1) dbo.(TESTTable1),100;g1) UPSERT:0%#:F dbo.(TESTTable1),100;g1	数	テール、選択 データベース ウィザードで作成されたSQL文。 SQL文を変更し、 ウィザードの結果を上巻きすることができます。
UPSERT(0)条件 dbo.(TESTTable1),[ID] = <1?(D?!>	CHITP.Body A Chiput.CSVilePath ClueBibb ClueBibb ClueScode CluerSMing CluerSM	列選択 □ 変換 UPSERTの条件 Join 文 サマリ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
		文換演SQLX
メイナミックマッピングUPSERT文の例: MERGE INTO MyTable USINGMyName= ?MyName?! WHEN NOT 合計	教しない条件 THEN INSERT WHEN 会教する条件 THEN UPDATE	
	次へ 終了 キャンセル	データベースウィザードにより牛成されたSOL文

T部のUPSERTの条件式を設定します。 下部のUPSERT条件欄のカーソルの位置を確認し、使用可能列の リストボックスから、テーブルの列をクリックして選択し、 [Enter]キーを押下すると、欄に列が入力補佐されます。 列をダブルクリックしても、欄に入力補佐されます。

= の右横に、<!?カラム名?!>と入力します。 この記入により、マッっピング画面でノードとして表示されます。 条件式の入力が完了したら、[次へ]をクリックします。

All Rights Reserved, Copyright(C) Magic Software Japan K.K. 2020

認・編集画面に遷移します。 SQL文の最終的な編集を行うことが出来ます。





2.3 データベースウィザードで生成されたSQL文の修正

データベースウィザードで生成されたSQL文について(Microsoft SQLServerの場合の例)
 MERGE文の構文との対比

生成された SQL 文

```
MERGE INTO dbo.[TESTTable1] USING (SELECT 1 [one]) AS
dummy([one]) ON dbo.[TESTTable1].[ID] = <!?ID?!>
WHEN NOT matched THEN
INSERT ( [ID],[String1],[Date1],[Time1],[Log1] ) VALUES
( <!?[ID]?!>,<!?[String1]?!>,<!?[Date1]?!>,<!?[Time1]?!>,
<!?[Log1]?!> )
WHEN matched THEN
UPDATE SET
[ID]=<!?[ID]?!>,[String1]=<!?[String1]?!>,[Date1]=<!?[Dat
e1]?!>,[Time1]=<!?[Time1]?!>,[Log1]=<!?[Log1]?!>
```

MERGE文の構文

副表はダミーテーブルを使用。

(送り元のタイプがFlatFileや他のデータベースのテーブルなどに対処が可能)

条件はON句で主表の主キー列との条件式を構成する。(<!?[カラム名]?!>を記述する)



※最終的にこの構造パターンに組み立てるのは、Microsoft SQLServerも、Oracleも、DB2/400も同様です。

MERGE INTO 主表 USING 副表 ON (条件) WHEN NOT MATCHED THEN INSERT (列1 [, 列2 ...]) VALUES (値1 [, 値2 ...]) WHEN MATCHED THEN UPDATE SET 列1 = 値1 [, 列2 = 値2 ...]

- SQL文の修正 (Microsoft SQLServerの場合)

MERGE INTO [TESTTable1] USING (SELECT 1 [one]) AS dummy([one]) ON [TESTTable1].[ID] = <!?ID?!> WHEN NOT matched THEN

INSERT ([ID],[String1],[Date1],[Time1],[Log1]) VALUES
(<!?[ID]?!>,<!?[String1]?!>,<!?[Date1]?!>,<!?[Time1]?!>,<!?[Log1]?!>)

WHEN matched THEN

UPDATE SET [String1]=<!?[String1]?!>,[Date1]=<!?[Date1]?!>,[Time1]=<!?[Time1]?!>,[Log1]=<!?[Log 1]?!>

UPDATE文のSET箇所の主キーカラムの削除(主キーを更新しない:[ID])。 必要に応じて、スキーマ名の省略(「dbo.」の削除)。

省略すると、一般的にはログオンユーザと同じスキーマまたはデフォルトのスキーマが適用されます。



- SQL文の修正 (Oracleの場合)

MERGE INTO XPI."TESTTABLE1" USING SYS.dual ON (XPI."TESTTABLE1"."ID"= <!?ID?!>) WHEN MATCHED THEN

UPDATE SET "STR1"=<!?"STR1"?!>,"DATE1"=<!?"DATE1"?!>,"LOG1"=<!?"LOG1"?!>

WHEN NOT MATCHED THEN

INSERT ("ID","STR1","DATE1","LOG1") VALUES
(<!?"ID"?!>,<!?"STR1"?!>,<!?"DATE1"?!>,<!?"LOG1"?!>)

UPDATE文のSET箇所の主キーカラムの削除(主キーを更新しない:"ID")。 必要に応じて、スキーマ名の省略(「XPI.」の削除 / SYS.dual の「SYS.」も削除(省略)してもOK)。 省略すると、一般的にはログオンユーザと同じスキーマまたはデフォルトのスキーマが適用されます。



 SQL文の修正(DB2/400の場合) データベースウィザードで生成されるSQL文の構文パターン MERGE INTO <ライブラリ名>.<テーブル名(ファイル名) > AS tgt USING (SELECT [カラム], [カラム…] FROM <ライブラリ名>.<テーブル名(ファイル名) >) AS src ON (tgt.<主キーとなるカラム名> = <!?[カラム名]?!>) WHEN MATCHED THEN UPDATE SET [カラム名]=<!?[カラム名]?!> [, カラム名=<!?カラム名?!>…] WHEN NOT MATCHED THEN INSERT ([カラム名][, カラム名…]) VALUES (<!?[カラム名]?!> [,<!?カラム名?!>…])

USING の副表と主表が同じであるため、DB2/400では、実行時にエラーとなることが確認されています。 USINGはダミーテーブルを利用するよう修正します。

修正例1: VALUES('DUMMY')

修正例 2 : SELECT * FROM (VALUES('DUMMY')) AS T1

(注:SELECTでVALUESを扱う場合、AS句で別名を指定する必要があります。指定しないとエラーとなります。)

UPDATE文のSET箇所の主キーカラムの削除(主キーを更新しない)。



- 修正する場合のその他の留意点
 - ON句の条件で使用するカラムは、主キーはユニークキーまたはプライムキーであること。
 - 主キーが複合主キー(複数のカラムの組み合わせで構成)の場合は、ON句にすべてのカラムの条件式を記述する

([複合主キーカラム1] = <!?[カラム1]?!>, [複合主キーカラム2] = <!?[カラム2]?!> [,…])



THANK YOU!



