

Přehled užitečných databázových funkcí

Seznámení s řešením pro zaznamenávání změn
v databázích MS SQL a PostgreSQL

- Potřeba uchování předchozích záznamů v databázi GENASIS (MS SQL)

- Obecně použitelné v jiných databázových aplikacích
- Co nejméně závislé na aplikační vrstvě

- Uložené procedury a funkce
- T-SQL
- Primární databáze a k ní vytvořená stínová databáze s historií (sufix `_hist`)

- Nutná existence primárního nebo unikátního klíče (identity atribut)
- Nutná existence atributů dateCreate, dateState (datetime), idAuthorCreate a idAuthor (int) - metadata
- Nastavení triggerů pro INSERT, UPDATE a DELETE

- Vytváří se automaticky při prvním použití triggeru
- Vlastní primární klíč idHist
- Původní identity atribut zaindexován
- Přidaný atribut deleted

- Ruční nastavení triggerů na tabulkách
- INSERT - vyplnit `idAuthorCreate`
- UPDATE - vyplnit `idAuthor`
- DELETE - použít proceduru `sp_Delete`

- Uložená procedura:

sp_Delete tableName varchar, idAuthor int, filter varchar, extended bit

- Základní:

EXEC sp_Delete 'MyTable', 2, 'id=10', 0

- Rozšířený:

EXEC sp_Delete 'm', 2, 'FROM MyTable m, OtherTable o WHERE m.id=o.myId', 1

- Vyšší komfort použití díky lepší programovatelnosti (plpgsql)
- Použití funkcí zabalených do samostatného modulu rozšíření `pg_hist`
- Primární schéma a k němu vytvořené stínové schéma (sufix `_hist`)

- Soubory ve sdíleném adresáři *extension*
- Příkaz:

```
CREATE EXTENSION pg_hist
```

- Nutná existence primárního nebo unikátního klíče i složeného - identity atribut(y)
- Nastavení sledování historie pomocí funkce `pgh_add_table_history`
- Odstranění sledování historie pomocí funkce `pgh_remove_table_history`

- Syntaxe:

```
pgh_add_table_history(table_name text  
[, schema_name text])
```

- Vytvoří metadatové atributy
userCreated a userChanged (integer)
dateCreated a dateChanged (timestamp)
- Nastaví triggery pro INSERT, UPDATE a DELETE

- Syntaxe:

```
pgh_remove_table_history(table_name text  
[, schema_name text])
```

- Odstraní metadatové atributy
**userCreated, userChanged,
dateCreated, dateChanged**
- Odstraní triggerů pro **INSERT, UPDATE a
DELETE**

- Vytvoří se automaticky při prvním použití UPDATE
- Sufix `_hist`
- Vlastní primární klíč `id_hist`
- Původní identity atribut(y) zaindexovány
- Přidaný atribut `deleted`

- Při změně struktury primární tabulky dojde při prvním použití ke změně stínové tabulky
- Odstranění sloupce nemá vliv
- Přejmenování vytvoří nový
- Změna datového typu - přejmenování původního (časové razítko) a vytvoření nového

- INSERT - vyplnit userCreated
- UPDATE - vyplnit userChanged
- DELETE - použít funkci pgh_delete

- Uživatelská funkce:

```
pgh_delete(user_id integer, table_name text [,
condition text [, using_clause text] ])
```

- Jednoduchý:

```
SELECT pgh_delete_(3, 'product', 'endDate IS NOT
NULL')
```

- Složený („join“)

```
SELECT pgh_delete(3, 'product p',
'p.idProduct=pp.idProduct AND pp.idProducer=5',
'producerHasProduct AS pp')
```

	MS SQL	PostgreSQL
Jazyk	Transact-SQL	Plpgsql
Programovatelnost	Nedostatečná, komplikované konstrukce	Velice dobrá
Možnosti implementace	Procedury a funkce	Funkce v uceleném modulu rozšíření
Instalace do databáze	Manuální kopírování definice funkcí a procedur	Snadná instalace rozšíření
Použití na tabulce	Manuální vytvoření potřebných atributů a nastavení triggerů	Snadná aplikace na tabulku
Identity atribut	Autoincrement i složený PK	PK nebo Unique (i složený)
Změna struktury	Ruční zásah	Automatická synchronizace