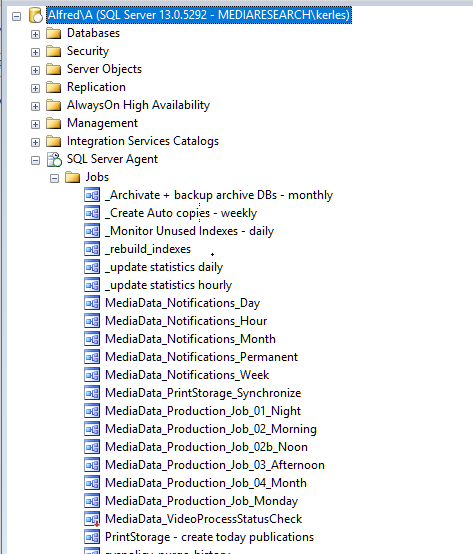
Slovnik pojmu

# OOH (patri sem outdoor)

OutOfHome zahrnuje OOHTv a Outdoor (moznosti v comboboxu v Kodovadle)

# Job na serveru:

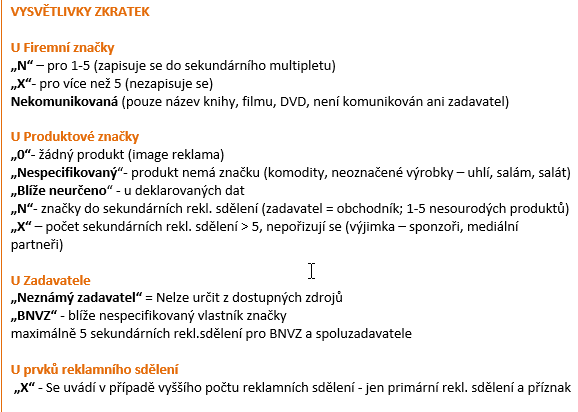


# Programovy blok na TvMedia messagi

TvMM ma ProgrammeBlockId. Potrebujeme vedet pocty sponzoringu k celemu programu. Hokej ma x reklamnich bloku. Cely hokej, je programovy blok. Napocitany start a end , proste si tam napocitavam veci, ktere jsou spolecne pro tech 5 MM. Ja nebudu Start a End bloku davat na kazdou z tech messagi. Je to ciste relacni zalezitost.

Programovy blok ma ted stringy – programmeBefore, programmeAfter atd. Algoritmus ktery pocita porady pred a po veme atomy, atomy musi dat 24 hodin.. Ta message se bafne a rekne se: Porad pred je 1. Tretina, do after se napise 2.tretina a jeste je porad in a tam se napise 2. Tretina. V BG se ceni podle poradu.

# Kodovadlo – vysvetleni zkratek pro brandy - N, X, 0, Blize neurceno, nespecifikovany, nekomunikovana …



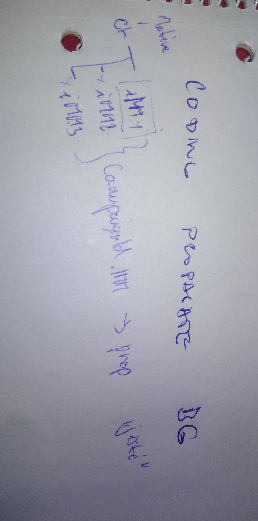
# CMP = CreativeMatchingProcess (na CreativeItem)

# CMS = CreativeMatchingStatus (na CreativeItem)

# Propagace

**–** Mame tri internet MM ktere jsou pres creativu provazane s motivem. Vsechny tri MM maji stejnou kreativu. Na InternetMM (v BG) mame sloupec CampaignId. Do K. nacitame jen jednu, holka ji okoduje, pri save se zavola propagace, ta si nacte ty zbyle dve a nastavi jim plausibility sure. Jelikoz pres kreativu sdili vsechny motiv tak jsou vsechny okodovane.

V cechach se propaguje jinak.



Pri savu internetu jestli se vola coding propagate storovka. Vola se ze dvou mist I dvoutransakcne.

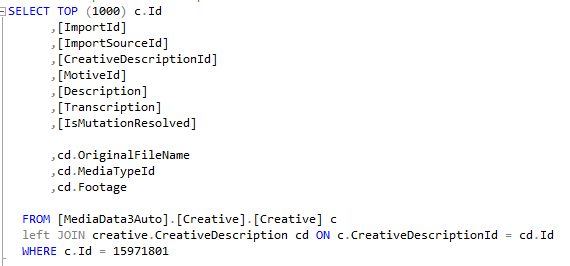
# Transkripce

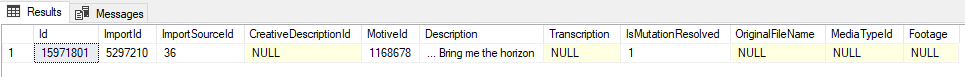
**– preklad description -**  na tabulce Creative.Creative, pouziva se jen na BG

# Description

**–** popis kreativy, najdeme na tabulce Creative.Creative. Tabulka Creative.Creative ma na sobe CreativeDescriptionId (muze byt null) kterym muzeme prijoinovat jeste tabuli CreativeDescription, kde jsou sloupce jako OriginalFileName, nebo CreativeDecriptionStatus, MediaTypeId a Footage.

V Normovadle – porovnani kreativ, zadam NormCreativeId = 15971801. To je Id normy a zaroven kreativy.





# CreativeDescriptionStatus

– ciselnik - muze byt Novy nebo Zpracovany

**Transkription**

# Rating poradu

**-**  kolik lidi se divalo

# SlicedChannel

– rozdeleny kanal napr CT-Decko a odpoledne Art

BulkInsert neboli Insert batch v linq

BATCHSIZE =batch\_size  
Specifies the number of rows in a batch. Each batch is copied to the server as one transaction. If this fails, SQL Server commits or rolls back the transaction for every batch. By default, all data in the specified data file is one batch. For information about performance considerations, see "Remarks," later in this topic.

# **Closure – uzaver, Disposed closure**

**Mame nejakou promennou ktera je mimo rozsah lambda vyrazu. Lambda vyrazy maji tzv odlozenou exekuci. Pokud je tato promenna objekt ktery se da disposnout, dostanu od Resharperu varovani na disposed closure.**

<https://www.jetbrains.com/help/resharper/AccessToDisposedClosure.html>

First of all, let's make sure that you understand what a [closure](https://en.wikipedia.org/wiki/Closure_(computer_programming)) is. To put it simply, a closure in C# is a lambda expression or an anonymous method that captures some variables from an outer scope. Here is the simplest example:

// A self-contained lambda. Not a closure.

Action printOne = () => { Console.WriteLine("one"); };

// A closure - a lambda that captures a variable from an outer scope.

string myStr = "one";

Action print = () => { Console.WriteLine(myStr); };