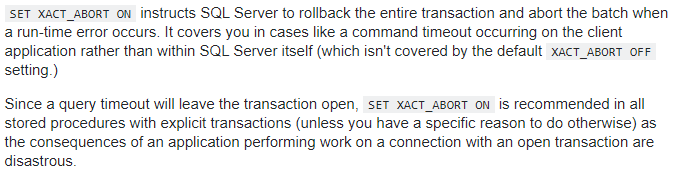
SQL TRAINING

### Identity insert

<https://www.tsql.info/set/set-identity-insert.php>

Po zapnuti muzu insertovat s presne urcenym Id i doprostred tabulky.

### SET XACT\_ABORT ON



|  |
| --- |
| CREATE PROCEDURE TransactionTest  AS      SET XACT\_ABORT ON      BEGIN TRANSACTION            -- Korektní SQL statement uvnitř transakce          INSERT INTO TransactionTestTable(TransactionName) VALUES('První insert')            -- Chybný SQL statement uvnitř transakce, do identity-column nelze zapisovat          INSERT INTO TransactionTestTable(TransactionTestID, TransactionName) VALUES(1, 'Druhý insert')            -- Korektní SQL statement uvnitř transakce, po chybě          INSERT INTO TransactionTestTable(TransactionName) VALUES('Třetí insert')        COMMIT TRANSACTION  GO |

### Proc je select query v transakci ?

A jeste se chci zeptat: Je tam transakce read uncommited, ve ktere je jen select query. Google rika, ze to ma vyznam jako NO LOCK. Tj forcene to issolation level a ja muzu cist data, ktera by jinak mohla byt zamcena. Chapu to spravne ?



### OPTION (FORCE ORDER)

FORCE ORDER in SQLServer. The FORCE ORDER is a query hint it executes the order of the tables exactly specified in a statement. When we use this query hint in a statement it will tellSQL server not to change the order of the joins in the query

Priklad je propagacni query v Kodovadle. Bez option trva 37 minut, s option jednu sekundu. Dodrzuje se poradi joinu, proto server v exekucnim planu nejde po indexech a muze mit mene radku k rozhodovani. Kdyz Option zakomentuju tak nedobehne:

exec sp\_executesql N'SELECT message.\*

FROM

[Media].[MediaMessage] [message]

INNER JOIN [Creative].[Creative] [creative] ON [message].[NormCreativeId] = [creative].[Id]

INNER JOIN [Media].[MotiveVersion] [motiveVersion] ON [creative].[MotiveId] = [motiveVersion].[MotiveId]

WHERE

[motiveVersion].[Id] = @p1 AND [motiveVersion].[ActiveFrom] <= [message].[AdvertisedFrom] AND

[motiveVersion].[ActiveTo] > [message].[AdvertisedFrom] and

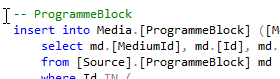
[message].[NormCreativeId] = @p2

OPTION (FORCE ORDER)

',N'@p1 int,@p2 int',@p1=1503113,@p2=16319083

### Synonyma

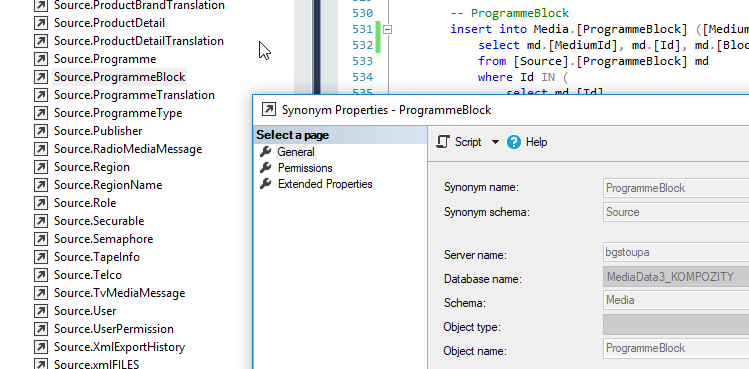
proc\_Codebooks\_Synchronize . V Adwindu budou taky BroadcastingId sloupce na tabulce ProgrammeBlock ? Kde se bere [Source] ? storovka nema zadny parametr.. Ze se to synchronizuje s Adwindem vim jen z nazvu

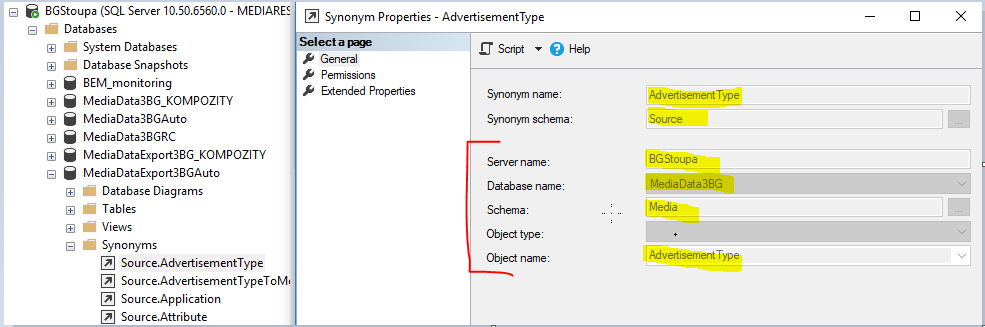


**Mira:**

Souce je název synonyma...k čemu slouží synonyma je jasné...je to něco jako "sql interface"...

- na každém serveru vedou do jiné source databáze

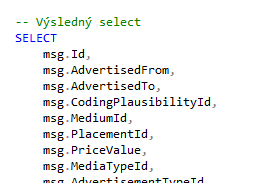




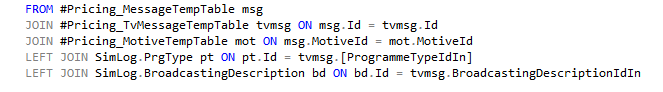
### Vysledny select, aneb kdyz je result storovky typ v .netu

proc\_PricingSelectTvMessagesByParameters

Procedura neco dela, tvori si temp tabule, v nich neco sasi a na konci vrati select, ktery nazvy sloupcu odpovida typu TvPricingItem v cenikovadle. (storovka se vola pomoci accessoru)



…



### Jak rozlisit jesty je result storovky typ v .netu nebo to jen nekdo vyjizdi do excelu

[‎21.‎10.‎2019 10:32] Peter Hlavenka:

jak poznam, jestli result storovky je nejaky typ v .netu, nebo si to jen nekdo vyjizdi do xlsx ? Treba Media.report\_kontrola\_nadlimitnich\_cen .

[‎21.‎10.‎2019 10:35] Miroslav Špaček:

1. nepoznáš, to prostě musíme vědět

když je to v Reports adresáři, tak je to zatím vždy excel

když je to ve storovkách

- a je **na konci select**, tak **appka/.net**

[‎21.‎10.‎2019 10:36] Miroslav Špaček:

- není na konci select, je to výkonná storovka jen na serveru

### Drop procedure – jak upravit storovku

Miroslav Špaček2 minutes ago

Na storovky nikdy nedělej changeScript, smaž jej...

Storovku edituj přímo v repu:

..\AdIntelGit\data\DatabaseModel\MediaData.Database\Create Scripts\Procedures\Import\proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert.sql

ať je vidět její historie změn...

odtud ji rovnou i nasazuj

if exists(select \*

from information\_schema.routines

where specific\_schema = N'Import'

and specific\_name = N'proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert'

)

drop procedure [Import].[proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert];

GO

/\*

Author: Miroslav Spacek

Created: 09.03.2011 - #12624. Skript na vložení TvLogu z CCC

Description: - máme skript, který vloží logy na základě přesahu a výjimek k nevložení

- chceme skript, který vloží pevně daná idéčka s jakýmkoliv přesahem (budou opravena v Z/N)

Updated: 09.02.2012, Miroslav Spacek - #16339. Rating u insertu MM at media

21.09.2012, Petr Dobes - #17647 Statistiky - storovka Import.proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert musí do Logging.Stat insertovat, pokud přiřadí motiv

06.09.2017, Ondrej Sadilek - #40930 Normovadlo - Porovnání záchytu a TvLogu - Dovolit vložit MM bez TapeCode

13.09.2017, Ondrej Sadilek - #40937 Normovadlo - Rozchodit Porovnání záchytu a TvLogů pro BG

02.05.2018, Hobbys - optimalizace nalezeni existujicich PrgBloku

11.05.2018, Kerles - vyplneni footage i v pripade, ze pod TapeCodem jeste neni zadna jina MM

14.05.2018, Kerles - logging

31.05.2018, Kerles - pokud nenajde pro nejaky TapeCode normu, tak se prerusi a vrati TapeCode

29.06.2018, Kerles - nenapojovat Normy, ktere maji nastaveno IgnoreDuplicty

09.10.2018, Kerles - pri vytvareni MM odecist TVLogShift od AdvFrom a AdvTo

08.03.2019, Peter Hlavenka - TFS 50464 - zmena znamenka TVLogShift - pri vytvareni MM pricist TVLogShift od AdvFrom a AdvTo

\*/

create procedure [Import].[proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert]

(

@InsertIds varchar(max) = '',

@InsertNormId int = '',

@ErrorTapeCode varchar(max) out

)

as

begin

--begin tran;

set nocount on;

-- Logging

declare @date datetime, @hostName varchar(100), @user varchar(50), @message varchar(max), @rowcount int, @duration float;

select @date = getdate(), @hostName = host\_name(), @user = suser\_name(), @message = ' - script proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert';

exec Logging.proc\_Log\_Save @Date = @date, @SessionId = '', @HostName = @hostName, @UserName = @user, @LogLevel = 'DEBUG', @Logger = 'proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert', @Thread = '', @Message = @message, @Exception = '';

declare @ErrorSave int, @ModifiedBy int, @Modified datetime, @IsMotiveFinded varchar(500), @HintId int, @CodingPlausibilityId int, @NewMMId int;

select @ModifiedBy = 220 /\*Import zpráv podle Tv logů\*/, @Modified = getdate();

-- vkladej pouze tyto TvLogy (definovano uzivatelem)

if object\_id('tempdb..#InsertIdsTable') is not null

drop table [dbo].[#InsertIdsTable]

create table [dbo].[#InsertIdsTable] (

[TvImportItemId] int NOT NULL primary key

) on [PRIMARY]

if (ltrim(rtrim(@InsertIds)) != '')

begin

declare @sql nvarchar(max)

set @sql='insert into #InsertIdsTable

select Id from Import.TvImportItem where Id in ('+@InsertIds+');';

exec(@sql)

end;

select @rowcount = @@rowcount, @duration = datediff(millisecond, @date, getdate()), @date = getdate(), @message = 'The number of inserted records into #InsertIdsTable: ' + ISNULL(CAST(@rowcount as varchar),'NULL');

exec Logging.proc\_Log\_Save @Date = @date, @SessionId = '', @HostName = @hostName, @UserName = @user, @LogLevel = 'DEBUG', @Logger = 'proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert', @Thread = '', @Message = @message, @Exception = '', @DurationInMiliseconds = @duration;

--select TvImportItemId as InsertIds from #InsertIdsTable order by TvImportItemId

-- najdi jim normu a motiv

create table #TvLogsForInsert (TvImportItemId int not null primary key, MotiveId int, NormCrId int, Footage int);

insert into #TvLogsForInsert

select tii.Id as TiiId,

(CASE WHEN @InsertNormId IS NOT NULL THEN

(select top 1 A.MotiveId from

(

select c.MotiveId, count(mm.Id) as pocet from Media.MediaMessage mm

inner join Media.TvMediaMessage tmm on mm.Id = tmm.Id

inner join [Creative].[Creative] c on mm.[NormCreativeId] = c.id

where mm.NormCreativeId = @InsertNormId

group by c.MotiveId

) A

ORDER BY A.pocet DESC)

ELSE

(select top 1 A.MotiveId from

(

select ti.TapeCode, c.MotiveId, count(mm.Id) as pocet from Media.MediaMessage mm

inner join Media.TvMediaMessage tmm on mm.Id = tmm.Id

inner join [Creative].[Creative] c on mm.[NormCreativeId] = c.id

inner join [Media].[TapeInfo] ti on [tmm].[TapeInfoId] = [ti].[Id]

where ti.TapeCode = tii.TapeCode

group by ti.TapeCode, c.MotiveId

) A

where A.TapeCode = tii.TapeCode

order by A.TapeCode, A.pocet desc) END) as MotiveId,

(CASE WHEN @InsertNormId IS NOT NULL THEN @InsertNormId ELSE(select top 1 B.NormCreativeId from

(

select ti.TapeCode, mm.NormCreativeId, count(mm.Id) as pocet from Media.MediaMessage mm

join Media.TvMediaMessage tmm on mm.Id = tmm.Id

inner join Creative.CreativeToCreativeItem ctci on ctci.CreativeId = mm.NormCreativeId

inner join Creative.CreativeItem ci on ci.Id = ctci.CreativeItemId

inner join [Media].[TapeInfo] ti on [tmm].[TapeInfoId] = [ti].[Id]

where ti.TapeCode = tii.TapeCode

and ci.IgnoreDuplicity = 0

group by ti.TapeCode, mm.NormCreativeId

) B

where B.TapeCode = tii.TapeCode

order by B.TapeCode, B.pocet desc) END) AS NormCrId,

(CASE WHEN @InsertNormId IS NOT NULL THEN

(SELECT TOP 1 C.Footage FROM

(SELECT COUNT(mm.Id) AS Pocet, tmm.Footage

FROM Media.MediaMessage mm

JOIN Media.TvMediaMessage tmm ON mm.Id = tmm.Id

JOIN Creative.Creative c ON mm.NormCreativeId = c.Id

WHERE mm.NormCreativeId = @InsertNormId

GROUP BY tmm.Footage) C

ORDER BY C.Pocet DESC)

ELSE

(SELECT COALESCE(

(SELECT top 1 C.Footage from

(SELECT ti.TapeCode, tmm.Footage, count(mm.Id) as pocet

FROM Media.MediaMessage mm

JOIN Media.TvMediaMessage tmm on mm.Id = tmm.Id

INNER join [Media].[TapeInfo] ti on [tmm].[TapeInfoId] = [ti].[Id]

WHERE ti.TapeCode = tii.TapeCode

GROUP by ti.TapeCode, tmm.Footage) C

WHERE C.TapeCode = tii.TapeCode

ORDER by C.TapeCode, C.pocet DESC),

DATEDIFF(SECOND,tii.AdvertisedFrom, tii.AdvertisedTo)))

END) as Footage

from Import.TvImportItem tii

join #InsertIdsTable iit on iit.TvImportItemId = tii.Id;

select @rowcount = @@rowcount, @duration = datediff(millisecond, @date, getdate()), @date = getdate(), @message = 'The number of inserted records into #TvLogsForInsert: ' + ISNULL(cast(@rowcount as varchar),'NULL');

EXEC Logging.proc\_Log\_Save @Date = @date, @SessionId = '', @HostName = @hostName, @UserName = @user, @LogLevel = 'DEBUG', @Logger = 'proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert', @Thread = '', @Message = @message, @Exception = '', @DurationInMiliseconds = @duration;

--select \* from #TvLogsForInsert;

-- jdi insertovat

declare @TvImportItemId int, @MotiveId int, @NormCreativeId int, @Footage int;

-- CURSOR

DECLARE vendor\_cursor CURSOR FOR

select TvImportItemId, MotiveId, NormCrId, Footage from #TvLogsForInsert;

OPEN vendor\_cursor

FETCH NEXT FROM vendor\_cursor

INTO @TvImportItemId, @MotiveId, @NormCreativeId, @Footage

WHILE @@FETCH\_STATUS = 0

begin

/\* neres konflikty

if not exists( -- konflikt s jiz existujici zpravou

select top 1 1 --@TvImportItemId, @MotiveId, mm.Id, mm.MotiveId

from Media.MediaMessage mm

where @MediumId = mm.MediumId

and (

@AdvFrom <= dateadd(second, -@ToleranceInSec, mm.AdvertisedTo)

and @AdvdTo >= dateadd(second, @ToleranceInSec, mm.AdvertisedFrom)

)

)

begin\*/

begin tran

select @duration = datediff(millisecond, @date, getdate()), @date = getdate(), @message = 'Inserting new message for TVImportItemId=' + ISNULL(cast(@TvImportItemId as varchar),'NULL');

EXEC Logging.proc\_Log\_Save @Date = @date, @SessionId = '', @HostName = @hostName, @UserName = @user, @LogLevel = 'DEBUG', @Logger = 'proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert', @Thread = '', @Message = @message, @Exception = '', @DurationInMiliseconds = @duration;

if (@MotiveId is null)

select @IsMotiveFinded = 'Nenalezen motiv', @CodingPlausibilityId = 1 /\*nezpracovane\*/

else

begin

select @IsMotiveFinded = 'Nalezen motiv', @CodingPlausibilityId = 8 /\*jiste\*/;

insert into Logging.Stat( [Key] ,Target ,TargetTypeId , Value1 ,Value1TypeId ,Value2 , Value2TypeId , Date , UserId)

values ( 601 , @NewMMId , 1 , @MotiveId , 2 , NULL , NULL , @Modified , @ModifiedBy )

end

select @duration = datediff(millisecond, @date, getdate()), @date = getdate(), @message = 'Motive searching: ' + @IsMotiveFinded;

EXEC Logging.proc\_Log\_Save @Date = @date, @SessionId = '', @HostName = @hostName, @UserName = @user, @LogLevel = 'DEBUG', @Logger = 'proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert', @Thread = '', @Message = @message, @Exception = '', @DurationInMiliseconds = @duration;

insert into Media.CodingHintContext (AdvertiserCrn, AdvertiserName, AgencyCrn, AgencyName, CampaignName, CompanyBrandName, ProductBrandName, ProductCategoryCode, ProductCategoryName, ProductSpecification, MotiveName)

select null, tvii.AdvertiserName, null, tvii.AgencyName, tvii.TapeName, tvii.CompanyBrandName, tvii.AdvertisementName, tvii.ProductCode, tvii.ProductName, 'Zdroj Tvlogy', @IsMotiveFinded from Import.TvImportItem tvii where Id = @TvImportItemId;

select @HintId = scope\_identity();

select @duration = datediff(millisecond, @date, getdate()), @date = getdate(), @message = 'HinId=' + ISNULL(CAST(@HintId AS VARCHAR),'NULL') + ' inserted to Media.CodingHintContext.';

EXEC Logging.proc\_Log\_Save @Date = @date, @SessionId = '', @HostName = @hostName, @UserName = @user, @LogLevel = 'DEBUG', @Logger = 'proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert', @Thread = '', @Message = @message, @Exception = '', @DurationInMiliseconds = @duration;

if (@NormCreativeId is null)

begin

select @duration = datediff(millisecond, @date, getdate()), @date = getdate(), @message = 'No norm for TapeCode found in databse. Exitting with Result = 1.';

exec Logging.proc\_Log\_Save @Date = @date, @SessionId = '', @HostName = @hostName, @UserName = @user, @LogLevel = 'DEBUG', @Logger = 'proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert', @Thread = '', @Message = @message, @Exception = '', @DurationInMiliseconds = @duration;

select @ErrorTapeCode = tvii.TapeCode from Import.TvImportItem tvii where tvii.Id = @TvImportItemId;

commit tran;

return;

end;

select @duration = datediff(millisecond, @date, getdate()), @date = getdate(), @message = 'NormCreativeId is: ' + ISNULL(CAST(@NormCreativeId AS VARCHAR),'NULL');

EXEC Logging.proc\_Log\_Save @Date = @date, @SessionId = '', @HostName = @hostName, @UserName = @user, @LogLevel = 'DEBUG', @Logger = 'proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert', @Thread = '', @Message = @message, @Exception = '', @DurationInMiliseconds = @duration;

--select @NormCreativeId as normid;

insert into Media.MediaMessage (ImportId, MediaTypeId, AdvertisementTypeId, AdvertisementSourceId, AdvertisedFrom, AdvertisedTo, CodingHintContextId,

CodingPlausibilityId, Ready, MediumId, PriceCurrencyId, PriceScopeId, IsPriceUnknown,

NormCreativeId, [CreativeId], Created, CreatedBy)

select tvii.ImportId, 2, tvii.AdvertisementTypeId, 61, dateadd(millisecond, mv.TvLogShift, tvii.AdvertisedFrom), dateadd(millisecond, mv.TvLogShift, tvii.AdvertisedTo), @HintId,

@CodingPlausibilityId, 0, am.Id, 1, 1, 1, @NormCreativeId, @NormCreativeId, @Modified, @ModifiedBy

from Import.TvImportItem tvii

join Media.Medium am on am.OriginalId = tvii.MediumId

join Media.MediumVersion mv on mv.MediumId = am.Id and mv.ActiveFrom <= tvii.AdvertisedFrom and mv.ActiveTo > tvii.AdvertisedFrom

where tvii.Id = @TvImportItemId;

select @NewMMId = scope\_identity();

select @duration = datediff(millisecond, @date, getdate()), @date = getdate(), @message = 'New MediaMessageId=' + ISNULL(CAST(@NewMMId AS VARCHAR),'NULL') + ' inserted to Media.MediaMessage.';

EXEC Logging.proc\_Log\_Save @Date = @date, @SessionId = '', @HostName = @hostName, @UserName = @user, @LogLevel = 'DEBUG', @Logger = 'proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert', @Thread = '', @Message = @message, @Exception = '', @DurationInMiliseconds = @duration;

--select @NewMMId as mmId

-- propoj MM a logem - tim "vypnu" trigger (trigger jiz stejne neexistuje)

update Import.TvImportItem

set MediaMessageId = @NewMMId,

Modified = @Modified,

ModifiedBy = @ModifiedBy

where Id = @TvImportItemId;

select @duration = datediff(millisecond, @date, getdate()), @date = getdate(), @message = 'TVIMportItemId=' + ISNULL(CAST(@TvImportItemId AS VARCHAR),'NULL') + ' connected to MediaMessageId=' + ISNULL(CAST(@NewMMId AS VARCHAR),'NULL') + ' (inserted to Import.TvImportItem).';

EXEC Logging.proc\_Log\_Save @Date = @date, @SessionId = '', @HostName = @hostName, @UserName = @user, @LogLevel = 'DEBUG', @Logger = 'proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert', @Thread = '', @Message = @message, @Exception = '', @DurationInMiliseconds = @duration;

declare @TapeInfoId int, @ProgrammeBlockId int, @EmptyDateTime datetime;

select @TapeInfoId = null, @ProgrammeBlockId = null, @EmptyDateTime = null;

select @EmptyDateTime = '19990101';

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Prace s TapeInfo

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Kdyz dostanu normu, nemam TapeCode a nemohu insertovat TapeInfo. Bude tedy NULL, ktere lze insertovat do TvMediaMessage.\*/

IF(ISNULL(@InsertNormId, '') != '')

BEGIN

SELECT @TapeInfoId = NULL;

END

ELSE

BEGIN

insert into Media.TapeInfo(TapeAgencyName, TapeCode, TapeLength, TapeName)

select distinct tvii.AgencyName, tvii.TapeCode, tvii.TapeLength, tvii.TapeName

from Import.TvImportItem tvii

left join [Media].[TapeInfo] ti on

isnull(tvii.[TapeCode], '') = isnull(ti.[TapeCode], '')

and isnull(tvii.[TapeName], '') = isnull(ti.[TapeName], '')

and isnull(tvii.AgencyName, '') = isnull(ti.[TapeAgencyName], '')

and isnull(tvii.[TapeLength], -1) = isnull(ti.[TapeLength], -1)

where tvii.Id = @TvImportItemId and ti.Id IS null and isnull(tvii.TapeCode, '') != ''

if(@@ROWCOUNT > 1)

select @TapeInfoId = scope\_identity();

--select @TapeInfoId as ti

if (@TapeInfoId is null)

select @TapeInfoId = ti.[Id]

from Import.TvImportItem tvii

inner join [Media].[TapeInfo] ti on

isnull(tvii.[TapeCode], '') = isnull(ti.[TapeCode], '')

and isnull(tvii.[TapeName], '') = isnull(ti.[TapeName], '')

and isnull(tvii.AgencyName, '') = isnull(ti.[TapeAgencyName], '')

and isnull(tvii.[TapeLength], -1) = isnull(ti.[TapeLength], -1)

where tvii.Id = @TvImportItemId;

--select @TapeInfoId as ti

if (@TapeInfoId is null)

raiserror ('@TapeInfoId is null', 11, 11);

END;

select @duration = datediff(millisecond, @date, getdate()), @date = getdate(), @message = 'TapeInfoId=' + ISNULL(CAST(@TapeInfoId AS VARCHAR),'NULL');

EXEC Logging.proc\_Log\_Save @Date = @date, @SessionId = '', @HostName = @hostName, @UserName = @user, @LogLevel = 'DEBUG', @Logger = 'proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert', @Thread = '', @Message = @message, @Exception = '', @DurationInMiliseconds = @duration;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Prace s ProgrammeBlock

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

-- Chybejici programove bloky

INSERT INTO Media.ProgrammeBlock(MediumId, BlockIdent, BlockCode, BlockStart, BlockEnd, BlockUnits,

BlockRating, ProgrammeBefore, ProgrammeAfter, ProgrammeTypeIdBefore,

ProgrammeTypeIdAfter, [BlockLength])

SELECT DISTINCT am.Id, tvii.BlockIdent, tvii.BlockCode, tvii.BlockStart, tvii.BlockEnd, tvii.BlockUnits,

null, tvii.ProgrammeBefore, tvii.ProgrammeAfter, ptBefore.Id, ptAfter.Id, datediff(second, tvii.BlockStart, tvii.BlockEnd)

from Import.TvImportItem tvii

inner join Media.Medium am on am.OriginalId = tvii.MediumId

LEFT JOIN Media.ProgrammeType ptBefore ON tvii.ProgrammeTypeBefore = ptBefore.Name

LEFT JOIN Media.ProgrammeType ptAfter ON tvii.ProgrammeTypeAfter = ptAfter.Name

left outer hash join [Media].ProgrammeBlock pb ON

am.Id = pb.[MediumId]

AND ISNULL(tvii.BlockIdent, '') = ISNULL(pb.BlockIdent, '')

AND ISNULL(tvii.BlockCode, '') = ISNULL(pb.BlockCode, '')

AND ISNULL(tvii.BlockStart, @EmptyDateTime) = ISNULL(pb.BlockStart, @EmptyDateTime)

AND ISNULL(tvii.BlockEnd, @EmptyDateTime) = ISNULL(pb.BlockEnd, @EmptyDateTime)

AND ISNULL(tvii.BlockUnits, 255) = ISNULL(pb.BlockUnits, 255)

AND -1 = ISNULL(pb.BlockRating, -1)

AND ISNULL(tvii.ProgrammeBefore, '') = ISNULL(pb.ProgrammeBefore, '')

AND ISNULL(tvii.ProgrammeAfter, '') = ISNULL(pb.ProgrammeAfter, '')

AND ISNULL(ptBefore.Id, 255) = ISNULL(pb.ProgrammeTypeIdBefore, 255)

AND ISNULL(ptAfter.Id, 255) = ISNULL(pb.ProgrammeTypeIdAfter, 255)

AND -1 = ISNULL(pb.[BlockLength], -1)

where tvii.Id = @TvImportItemId and pb.Id IS NULL

and -- nezkladej prazdny

(

ISNULL(tvii.BlockIdent, '') != ''

OR ISNULL(tvii.BlockCode, '') != ''

OR ISNULL(tvii.BlockStart, @EmptyDateTime) != @EmptyDateTime

OR ISNULL(tvii.BlockEnd, @EmptyDateTime) != @EmptyDateTime

OR ISNULL(tvii.BlockUnits, -1) != -1

OR ISNULL(tvii.ProgrammeBefore, '') != ''

OR ISNULL(tvii.ProgrammeAfter, '') != ''

OR ISNULL(ptBefore.Id, 255) != 255

OR ISNULL(ptAfter.Id, 255) != 255

)

if(@@ROWCOUNT > 1)

select @ProgrammeBlockId = scope\_identity();

SELECT @duration = datediff(millisecond, @date, getdate()), @date = getdate();

IF (@ProgrammeBlockId is NOT NULL)

SELECT @message = 'ProgramBlock searching STEP 1 of 2. ProgrammeBlockId=' + ISNULL(CAST(@ProgrammeBlockId AS VARCHAR),'NULL');

ELSE

SELECT @message = 'ProgramBlock searching STEP 1 of 2 finished. @ProgrammeBlockId not found. Starting STEP 2: @TvImportItemId=' + ISNULL(CAST(@TvImportItemId AS VARCHAR),'NULL')

+ ', @EmptyDateTime=' + ISNULL(CAST(@EmptyDateTime AS VARCHAR),'NULL');

EXEC Logging.proc\_Log\_Save @Date = @date, @SessionId = '', @HostName = @hostName, @UserName = @user, @LogLevel = 'DEBUG', @Logger = 'proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert', @Thread = '', @Message = @message, @Exception = '', @DurationInMiliseconds = @duration;

--select @ProgrammeBlockId as pb

if (@ProgrammeBlockId is null)

SELECT @ProgrammeBlockId = pb.id

from Import.TvImportItem tvii

inner join Media.Medium am on am.OriginalId = tvii.MediumId

inner join [Media].ProgrammeBlock pb on

am.Id = pb.[MediumId]

AND ISNULL(tvii.BlockIdent, '') = ISNULL(pb.BlockIdent, '')

AND ISNULL(tvii.BlockCode, '') = ISNULL(pb.BlockCode, '')

AND ISNULL(tvii.BlockStart, @EmptyDateTime) = ISNULL(pb.BlockStart, @EmptyDateTime)

AND ISNULL(tvii.BlockEnd, @EmptyDateTime) = ISNULL(pb.BlockEnd, @EmptyDateTime)

AND ISNULL(tvii.BlockUnits, 255) = ISNULL(pb.BlockUnits, 255)

AND -1 = ISNULL(pb.BlockRating, -1)

AND ISNULL(tvii.ProgrammeBefore, '') = ISNULL(pb.ProgrammeBefore, '')

AND ISNULL(tvii.ProgrammeAfter, '') = ISNULL(pb.ProgrammeAfter, '')

AND ISNULL(cast(tvii.[ProgrammeTypeBeforeCode] as smallint), 255) = ISNULL(pb.ProgrammeTypeIdBefore, 255)

AND ISNULL(cast(tvii.[ProgrammeTypeAfterCode] as smallint), 255) = ISNULL(pb.ProgrammeTypeIdAfter, 255)

AND -1 = ISNULL(pb.[BlockLength], -1)

where tvii.Id = @TvImportItemId;

SELECT @duration = datediff(millisecond, @date, getdate()), @date = getdate();

IF (@ProgrammeBlockId is NOT NULL)

SELECT @message = 'ProgramBlock searching STEP 2 of 2 finished. @ProgrammeBlockId = ' + ISNULL(CAST(@ProgrammeBlockId AS VARCHAR),'NULL');

ELSE

SELECT @message = 'ProgramBlock searching STEP 2 of 2 finished. @ProgrammeBlockId not found.';

EXEC Logging.proc\_Log\_Save @Date = @date, @SessionId = '', @HostName = @hostName, @UserName = @user, @LogLevel = 'DEBUG', @Logger = 'proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert', @Thread = '', @Message = @message, @Exception = '', @DurationInMiliseconds = @duration;

--select @ProgrammeBlockId as pb

--Zakomentovana kontrola kvuli tasku nize

--#40937 Normovadlo - Rozchodit Porovnání záchytu a TvLogů pro BG

--if (@ProgrammeBlockId is null)

-- raiserror ('@ProgrammeBlockId is null', 11, 11);

--MM

insert into Media.TvMediaMessage(Id, Footage, Rating, TapeInfoId, ProgrammeBlockId, BlockPosition)

select @NewMMId, @Footage, null, @TapeInfoId, @ProgrammeBlockId, tvii.BlockPosition

from Import.TvImportItem tvii where Id = @TvImportItemId;

select @duration = datediff(millisecond, @date, getdate()), @date = getdate(), @message = 'New MediMessageId=' + ISNULL(CAST(@NewMMId AS VARCHAR),'NULL') + ' inserted into Media.TvMediaMessage.';

EXEC Logging.proc\_Log\_Save @Date = @date, @SessionId = '', @HostName = @hostName, @UserName = @user, @LogLevel = 'DEBUG', @Logger = 'proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert', @Thread = '', @Message = @message, @Exception = '', @DurationInMiliseconds = @duration;

commit tran

FETCH NEXT FROM vendor\_cursor

INTO @TvImportItemId, @MotiveId, @NormCreativeId, @Footage

END

CLOSE vendor\_cursor

DEALLOCATE vendor\_cursor

set nocount off;

--vypisy (pouziva Normovadlo pro informovani uzivatele o poctu zalozenych MM)

select @duration = datediff(millisecond, @date, getdate()), @date = getdate(), @message = 'Starting "Vypisy". @Modified =' + ISNULL(CAST(@Modified AS VARCHAR),'NULL');

EXEC Logging.proc\_Log\_Save @Date = @date, @SessionId = '', @HostName = @hostName, @UserName = @user, @LogLevel = 'DEBUG', @Logger = 'proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert', @Thread = '', @Message = @message, @Exception = '', @DurationInMiliseconds = @duration;

select tvii.id as TvImportItem, mm.Id as NewMediaMessageId, mm.MediumId, mm.CodingPlausibilityId, mm.AdvertisedFrom, mm.AdvertisedTo, pb.id as PrgBlockId, pb.[BlockStart], pb.[BlockEnd], c.MotiveId, mv.Id as MotiveVersionId, mm.NormCreativeId, mm.CodingHintContextId

from Media.MediaMessage mm

inner join [Media].[TvMediaMessage] tmm on mm.id = tmm.id

join Import.TvImportItem tvii on tvii.MediaMessageId=mm.Id

left join [Creative].[Creative] c on [mm].[NormCreativeId] = [c].[Id]

left join [Media].[MotiveVersion] mv on c.[MotiveId] = mv.[MotiveId] and mm.[AdvertisedFrom] >= mv.[ActiveFrom] and mm.[AdvertisedFrom] < mv.[ActiveTo]

left join [Media].[ProgrammeBlock] pb on tmm.[ProgrammeBlockId] = pb.id

where mm.Created = @Modified

order by mm.MediumId, mm.AdvertisedFrom;

select @duration = datediff(millisecond, @date, getdate()), @date = getdate(), @message = '"Vypisy" finished. Deleting tempdb...';

EXEC Logging.proc\_Log\_Save @Date = @date, @SessionId = '', @HostName = @hostName, @UserName = @user, @LogLevel = 'DEBUG', @Logger = 'proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert', @Thread = '', @Message = @message, @Exception = '', @DurationInMiliseconds = @duration;

if object\_id('tempdb..#InsertIdsTable') is not null

drop table [dbo].[#InsertIdsTable];

SELECT @duration = datediff(millisecond, @date, getdate()), @date = getdate(), @message = 'Tempdb deleted. Procedure ImportTvMediaMessageFromTvLogs finished.';

EXEC Logging.proc\_Log\_Save @Date = @date, @SessionId = '', @HostName = @hostName, @UserName = @user, @LogLevel = 'DEBUG', @Logger = 'proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert', @Thread = '', @Message = @message, @Exception = '', @DurationInMiliseconds = @duration;

--commit tran;

SET @ErrorTapeCode = '';

end;

GO

GRANT EXEC ON [Import].[proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert] TO MediaDataBasicAccess

/\*

SELECT top 100 \* FROM Import.TvImportItem where Id in (417405,420346);

exec Import.proc\_ImportTvMediaMessageFromTvLogs\_03\_Insert

@InsertIds='417405, 420346';

\*/

### prevedeni int poctu sekund na datetime sloupec

CONVERT(VARCHAR, DATEADD(second, VideoLength, 0),108) AS ValidLenght,



### tempdb

USE tempdb

create table Person

(

Id int not null Primary key,

Gender int null

)

-- DROP TABLE dbo.Person

### Vymazani cache Stoupovi

<https://www.mssqltips.com/sqlservertip/1360/clearing-cache-for-sql-server-performance-testing/>

pozor, ať to nepustíš na metatronu, ale on by se z toho časem dostal M.



USE [MediaData3Auto];

GO

CHECKPOINT;

GO

DBCC DROPCLEANBUFFERS;

GO

### Basics

|  |  |
| --- | --- |
| Create database | Create Database Sample1 |
| Rename database | Alter database Sample1 Modify Name Sample2 |
| Deleting database | Drop database Sample2 |

### Creating

|  |  |
| --- | --- |
| Creating table | create table Person  (  Id int IDENTITY not null Primary key,  Gender int null  )  create table tblGender  **(**  Id int not null Primary key,  Gender int null  ) |

### Insert

|  |  |
| --- | --- |
| Insert | insert into Person (Id, Name, GenderId) values (5, 'Rena', 2) |

### Delete

DELETE FROM *table\_name*  
WHERE *condition*;

Když je join musim za delete definovat z ceho se ma mazat:

delete up from Membership.UserPermission up

join Membership.Securable sec on up.SecurableId = sec.Id

join Membership.Permission per on up.PermissionId = per.Id

where per.Code = 'Manage/execute'

and sec.Code = 'TvLogAdministrationViewModel'

delete stp from Membership.SecurableToPermission stp

join Membership.Securable sec on stp.SecurableId = sec.Id

join Membership.Permission per on stp.PermissionId = per.Id

where per.Code = 'Manage/execute'

and sec.Code = 'TvLogAdministrationViewModel'

### ****Constraints****

Rekneme jakou table chceme upravvovat – **alter table tableName**

Rekneme, ze pridavame constraint na tabuli Person a sloupec GenderId a ze to bude cizi klic **add constraint Person\_GenderId\_FK**

Rekneme typ omezeni a na jaky bude sloupec **Foreign Key (GenderId)** , ktery odkazuje na tabuli tblDender(sloupec Id) **references tblGender(Id)**

|  |  |
| --- | --- |
| **Add Constraints** | Alter table Person add constraint Person\_GenderId\_FK  Foreign Key (GenderId) references tblGender(Id) |
| **Default constraint** | Alter table Person add constraint DF\_Person\_GenderId  default 3 for GenderId  **[[1]](#endnote-1)** |
| **Odstranenei omezeni** | Alter table Person Drop constraint DF\_Person\_GenderId |
|  |  |
| **Check constraint** | Muzeme rict, ze napr do sloupce Age muzeme vkladat jen hodnoty Age > 0 and Age je <150 |



[10:34](https://www.youtube.com/watch?v=dwSqHhMl32Y)

### Update

UPDATE table\_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;

### Update tabulky z jine najoinovane tabulky tabulky

Zachytavadlo, analyza query. Mam tempTabuli a potrebuju joinout ctci. Z ctci pak chci updatnout sloupec v temp

<https://stackoverflow.com/questions/1604091/update-a-table-using-join-in-sql-server>

begin tran

UPDATE a

SET a.CalculatedColumn= b.[Calculated Column]

FROM table1 a INNER JOIN table2 b ON a.commonfield = b.[common field]

WHERE a.BatchNO = '110'

commit tran

UPDATE tcm set

tcm.CreativeItemId = ctci.CreativeItemId

from ##TempCatchingMessages as tcm

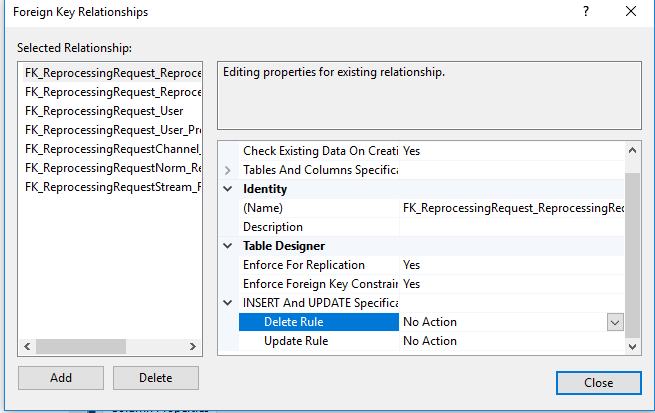
JOIN Creative.CreativeToCreativeItem ctci on tcm.CreativeId = ctci.CreativeId

### [Adding a default constraint - Part 4](https://www.youtube.com/watch?v=dwSqHhMl32Y)

[kudvenkat](https://www.youtube.com/watch?v=dwSqHhMl32Y)

[480 tis. zhlédnutí](https://www.youtube.com/watch?v=dwSqHhMl32Y)

**U omezeni je mozne definovat co ma SQL Server delat v pripade, ze je odstanena hodnota v tabulce kam ukazu je Foreign Key. Například v tabulce Persons mame osobu Jan , GenderId = 1 . Když z tabulky Gender odstranime radek Id = 1, dostaneme chybu, protože na toto Id ukazuje cizi klic. To je defaultni chovani. Další nastaveni jsou : Cascade (provede se i na navazanych tabulkach (smazani smaze i radky v jinych tabulkach kde cizi klice ukazuji na smazanou hodnotu), Set Nulls nastavi NULL tam kde FK ukazoval na smazanou hodnotu.**



**Pridani sloupce**

ALTER TABLE table\_name  
ADD column\_name datatype;

Alter table Person add Age int not null

**Check Constraint**

<https://www.youtube.com/watch?v=9Zj5ODhv0b0>

**Prida kontrolu hodnoty na zaklade Expression. (vyrazu)**

**Například potrebujeme, aby do sloupce Age nebylo mozne zadat zapornou hodnotu a hodnotu vetsi nez 150.**

Alter table Person add constraint CK\_Person\_Age CHECK (Age > 0 AND Age < 150)

### ****Invalid column name****

0

My Problem is , i was unable to see Error Message (Red Underline) in MS SQL although the column name is invalid

Press CTRL + Shift + R Its refresh an intellisense

### ****Identity Column****

<https://www.youtube.com/watch?v=aOkFE6NLGCQ>

Identity muzeme nastavit kliknutim pravym tlacitkem na tabulku -> Design -> IdentitySpecification -> IsIdentity -> Yes. Increment urcuje jaky bude increment cisel radku.

Pro vlozeni zaznamu bez definovaneho Id musime pouzit prikaz: SET IDENTITY\_INSERT Person OFF.

Nyní muzeme vkladat bez Id INSERT INTO Person (‘Michal‘)

Vymazena Id radku uz nejdou znovu pouzit . Existuje moznost jak nastavit identity v tabulce aby zacinala zase od 1. Musí se vymazat všechny zaznamy a pouzit prikaz:

DCCB CHECKIDENT (Person, RESEED, 0)

Vytvoreni tabulky s Identitou

Create table Test1

(

Id int Identity (1,1), => Nastavi identitu , increment a seed

Value nvarchar(20)

)

### How to get the last identity column value

<https://www.youtube.com/watch?v=n1iwngG_zNY>

**Pro ziskani poslední hodnoty Id v tabulce s identitou pouzijeme prikazy**

SELECT SCOPE\_IDENTITY()

Nebo:

SELECT @@IDENTITY

-- Scope identity muzu pouzit jen v pripade, ze jsem do nejake tabulky insertoval. Pak mi vrati nejvyssi Id teto (naposledy pouzite) tabulky (pokud ma ta tabulka identitu). Tj. insertnul jsem Pricing a Scope\_identity mi vrati Id pricingu.

-- SimLog uz ale existuje a proto musim ziskat Id jinak (script na pridavani securable)

### Triggers – automaticke operace

<https://www.youtube.com/watch?v=n1iwngG_zNY>

V prikladu bude trigger který po kazdem insertu do tabulky Test1 automaticky vytvori radek v tabulce Test2

CREATE TRIGGER trForInsert ON Test1 FOR INSERT

AS

BEGIN

INSERT INTO Test2 VALUES ('YYYY')

END

Po executnuti pod tabulkou Test1 ve slozce Triggers pribyde trigger



* The @@identity function returns the last identity created in the same session.
* The scope\_identity() function returns the last identity created in the same session and the same scope.
* The ident\_current(name) returns the last identity created for a specific table or view in any session.
* The identity() function is not used to get an identity, it's used to create an identity in a select...into query.

The session is the database connection. The scope is the current query or the current stored procedure.

A situation where the scope\_identity() and the @@identity functions differ, is if you have a trigger on the table. If you have a query that inserts a record, causing the trigger to insert another record somewhere, the scope\_identity() function will return the identity created by the query, while the @@identity function will return the identity created by the trigger.

So, normally you would use the scope\_identity() function.

Rozdil mezi triggerem a procedurou je ten ze trigger je zavolan vždy když se splni podmínka, kdezto procedura se musí volat explicitne.

### Unique Key Constraint

<https://www.youtube.com/watch?v=oqrsfatxTYE>

Na zamezeni duplicitnich hodnot pouzivame Unique Key Constraint

ALTER TABLE Person ADD CONSTRAINT UQ\_Person\_Email UNIQUE (Email)

**Pridani unique key na tabulku Media.SelfPromotionDefinition**

….jo, to vypadá dobře, jen poznámka, pokud má Id sloupec IDENTITu, je dobré používat ten a nespoléhat na nějaké datum created co vyplňuje kdokoliv, ale detail, M.

**From:** Peter Hlavenka <[Peter.Hlavenka@admosphere.cz](mailto:Peter.Hlavenka@admosphere.cz)>   
**Sent:** Thursday, June 13, 2019 3:19 PM  
**To:** Miroslav Špaček <[Miroslav.Spacek@admosphere.cz](mailto:Miroslav.Spacek@admosphere.cz)>  
**Subject:** RE: Definice vlastní inzerce - duplicity

Prosim o kontrolu :

       ;

with ##SelfPromotionDefinitionCTE AS

(

       Select \*, ROW\_NUMBER () Over(Partition by Media.SelfPromotionDefinition.OwnerId, Media.SelfPromotionDefinition.MediumId, Media.SelfPromotionDefinition.ActiveFrom order by Media.SelfPromotionDefinition.Created) as RowNumber

       from  [Media].[SelfPromotionDefinition]

)

--SELECT \* FROM ##SelfPromotionDefinitionCTE WHERE ##SelfPromotionDefinitionCTE.RowNumber > 1

Delete from ##SelfPromotionDefinitionCTE where RowNumber > 1

ALTER TABLE [Media].[SelfPromotionDefinition] ADD  CONSTRAINT UQ\_OwnerId\_MediumId\_ActiveFrom UNIQUE NONCLUSTERED

(

       [OwnerId] ASC,

       [MediumId] ASC,

       [ActiveFrom] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, IGNORE\_DUP\_KEY = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

GO

Vyzkouseno na MediaData3Auto i s overenim, ze tam od kazde duplicity zustane jen jeden (nejstarsi) zaznam

P.

### All about Select

<https://www.youtube.com/watch?v=R9pXnHIFj_8>

Distinct

SELECT DISTINCT City FROM Person

Vybere zaznamy které jsou rozlisne (distinct) . Pokud vybereme vic sloupcu, vrati se nam rozlisne zaznamy posuzovane vzhledem ke dvoum sloupcum tj jako stejne se budou posuzovat radky které mají stejnou hodnotu pro každý posuzovany sloupec.

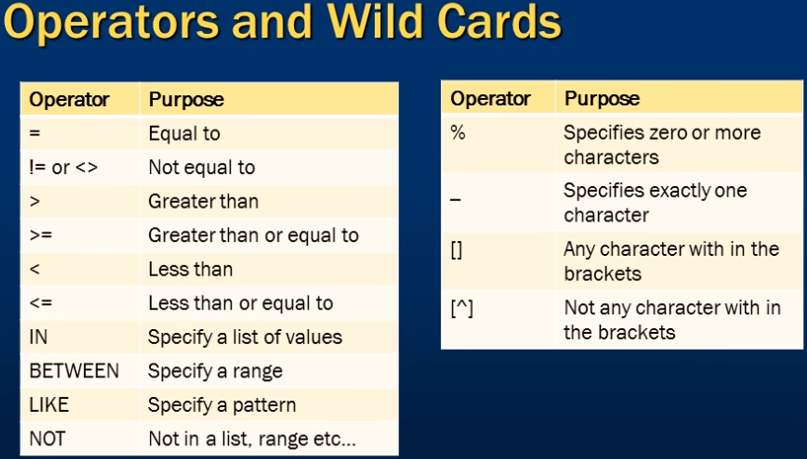
SELECT DISTINCT Name, City FROM Person

Not Equal

SELECT \* FROM Person WHERE City <> 'London'

Vybere zaznamy kde City není London . Operator <> stejne jako :

SELECT \* FROM Person WHERE City != 'London'



**Between**

SELECT \* FROM Person WHERE Age BETWEEN 8 AND 30

**Vrati zaznamy kde je vek mezi zadanou hodnotou vcetne zaznamu kde vek = 8 a 30**

**Like**

SELECT \* FROM PERSON WHERE CITY LIKE 'L%'

Vrati zaznamy které zacinaji na L. Nein CaseSensitive takze je mozne pouzit male l.

SELECT \* FROM PERSON WHERE Email LIKE '%@%'

Vrati zaznamy které mají nejake znaky před definovanym znakem (tady @ ) a nejake znaky po definovanem znaku (operator % ) (pozn. -> vrati validni emaily)

SELECT \* FROM PERSON WHERE Email **NOT** LIKE '%@%'

Not Like je obracena hodnota k like vrati zaznamy kde není uprostred zavinac.

SELECT \* FROM PERSON WHERE Email LIKE '\_@\_.com'

Podtrzitko znamena „Prave jeden znak“ Vrati emailove adresy kde je jeden znak před zavinacem a jeden po, plus .com

SELECT \* FROM PERSON WHERE Email LIKE '[MST]%'

Vrati emaily které zacinaji na pismena definavane v zavorkach Case insensitive (jako StartsWith)

SELECT \* FROM PERSON WHERE Name LIKE '[^PTO]%'

Operator ^ obrati vyraz , vrati ty které nezacinaji na tyto pismena

**Order By**

SELECT \* FROM PERSON ORDER BY NAME ASC , CITY DESC

Seradi vystup podle Name vzestupne a pokud je vice stejnych jmen, tyto zaznamy seradi ještě podle City sestupne

SELECT **TOP 10** \* FROM PERSON ORDER BY NAME ASC , CITY DESC

To same, ale vybere jen Top 10 radku

SELECT TOP 20 **PERCENT** \* FROM PERSON ORDER BY NAME ASC , CITY DESC

Vybere hornich 20 percent zaznamu

SELECT TOP 1 \* FROM PERSON ORDER BY AGE DESC

Vybere nejstarsi osobu

### Group By

<https://www.youtube.com/watch?v=FKSSOpQe5Jc>

**Agregovane funkce:**

SELECT SUM(SALARY) FROM PERSON

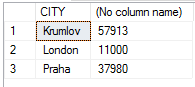
SELECT MAX(SALARY) FROM PERSON

SELECT AVG(SALARY) FROM PERSON

Group By pouziva jednu nebo vice agregovanych funkci.

SELECT CITY, SUM(SALARY) FROM PERSON GROUP BY CITY

Zjisti jaky je soucet platu pro kazde město:



### Joins

<https://www.youtube.com/watch?v=GKGtOABAO9s>

Cross Join => Každý radek z leve strany dostane radek z prave strany tj když ma leva tabulka 10 radku a prava 4 radky , tabulka která vznikne cross joinem bude mit 40 radku.

Self Join => Joinuje se tabulka sama se sebou



Advanced Joins:



SELECT ColumnList FROM LeftTable

JOIN RightTable ON LeftTable.RightTableId = RightTable.Id

**Left Join pouziju na spojeni dvou tabulek. Tabulky spojim na zakladne stejneho sloupce – Creative.Id == Message.CreativeId . V leve tabulce mam všechny hodnoty a v prave nektere chybi. Tam se dosadi null:**

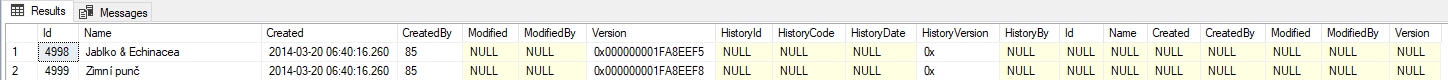
select \* from media.ProductDetail pro

left join History.ProductDetail his on pro.Id = his.Id

where his.Id is null

jinymi slovy : dej mi radky produktDetailu, ktere v history tabulce chybi

Hodnoty vlevo(left) existuji



### Select top 1 1

SELECT TOP 1 1 FROM [SomeTable] WHERE <SomeCondition> Means if the condition is true and any rows are returned from the select, only return top 1 row and only return integer 1 for the row (no data just the integer 1 is returned).

Pokud je result pravdivy, nebo by se vratily nejake radky kdybychom misto 1 1 pouzili 1 \*, vrati sql server pouze jeden radek kde bude jen jeden sloupec a v nom hodnota 0 nebo 1.

**Priklad:**

use MediaData3BGAuto

declare @mediumId tinyint, @dateTimeFrom datetime, @dateTimeTo datetime;

set @mediumId = '68'; --lze preskocit

--set @mediumId = '1'; --nelze preskocit

set @dateTimeFrom = '2019-04-12 01:00:00.000';

set @dateTimeTo = dateadd(hour, 1, @dateTimeFrom);

-- vrati jeden radek, vsechny sloupce

SELECT TOP 1 \*

FROM SimLog.BroadcastingDescription bd

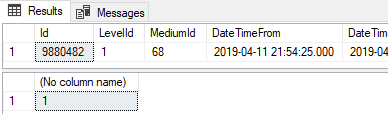
WHERE bd.MediumId IN (@mediumId) AND bd.DateTimeFrom > DATEADD(HOUR, -12, @dateTimeFrom) AND bd.DateTimeTo >= @dateTimeTo

-- vrati jen jeden radek a jeden sloupec s bit hodnotou

SELECT TOP 1 1

FROM SimLog.BroadcastingDescription bd

WHERE bd.MediumId IN (@mediumId) AND bd.DateTimeFrom > DATEADD(HOUR, -12, @dateTimeFrom) AND bd.DateTimeTo >= @dateTimeTo



### CASE Statement, Different way to replace NULL in SQL server, ISNULL

<https://www.youtube.com/watch?v=4ZoHY4RT1Fo>

Pokud nechceme aby sloupec zobrazoval hodnoty NULL ale místo toho vypsal neco smysluplnejsiho muzeme otestovat sloupect prikazem ISNULL a zamenit hodnotu v miste vyskytu Null za něco jineho.

Bohuzel nelze zmenit datovy typ takze pokud chci menit DepartmentId které je typu int muzu ho zamenit jen za hodnotu typu int tady 5000.

ISNULL([DepartmentId], 5000) as Department

**Lepsim prikladem je zamenit hodnotu ve sloupci jehož typ je nvarchar:**

SELECT ISNULL([ManagerName], ‘No Manager’) as Manager FROM dbo.Person

**Další moznost jak zmenit null hodnotu je pouzit CASE statement**

SELECT Name,

CASE

WHEN DepartmentId IS NULL

THEN 100 ELSE DepartmentId

END AS DepId

FROM dbo.Person

Vymeni DepartmentId v miste kde ma hodnotu null za hodnotu 100. Jedna se jen o zobrazeni dat v databazi se nic nemeni, porad je tam hodnota Null.

### Union and UnionAll

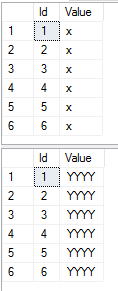
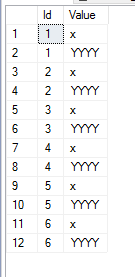
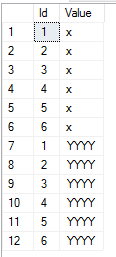
<https://www.youtube.com/watch?v=9w5uRCFOiTo>

Slouci dva a vice Selectu do jedne query. Union All vrati i duplicity kdezto Union ne. Poradi a počet promennych v selectech musí byt stejny.

select Id,Value from Test1 select Id,Value from Test1 select Id,Value from Test1

select Id,Value from Test2 UNION UNION ALL

select Id,Value from Test2 select Id,Value from Test2

### Storovane procedury

<https://www.youtube.com/watch?v=Qu3E-oncF3g>

Procedury jsou metody na serveru, slouzi k ulozeni částo pouzivanych operaci.

Vytvoreni

CREATE PROC sproc\_GetTestTables

AS

BEGIN

select Id,Value from Test1

select Id,Value from Test2

END

**Pouziti**

* Jmeno procedury oznacime a stlacime Execute button nebo F5
* Před jmeno dame klicove slovo exec nebo execute
* Nazev databaze -> Programmability => Stored Procedures -> Prave tlacitko mysi na procedure -> Execute stored procedure . Tady je dialogove okno ve kterem je mozne doplnit parametry pokud je procedura vyzaduje.

Vstupni parametry procedury

CREATE PROC spGetTables

@Gender nvarchar (20)

AS

BEGIN

INSERT INTO dbo.Test1 VALUES (@Gender)

END

Vice parametru spojujeme pomoci AND : Select \* from dbo.Person where Gender = @Gender AND DepartmentId = @DepId

**Volani s parametrem**

EXEC spSelectById 8

Jak se podivat na text jiz ulozene procedury

Nazev databaze -> Programmability => Stored Procedures -> Prave tlacitko mysi na procedure -> Script Stored Procedure As -> Create To -> New Query Editor

SQL Server ma systemove procedury. Oznacuji se **sp\_** Proto je lepsi pouzivat vlastní oznaceni procedur, aby nedoslo ke kolizi nazvu napr. v nove verzi serveru. (sproc\_ , spNazevMetody)

Jednou ze systemovych procedur je napr **sp\_helptext.**  Když za nej napiseme nazev procedury vypise jeji zneni. Za **sp\_help** muzeme napsat nazev procedury nebo tabulky dozvime se spoustu zajimavych informaci.

Na zjisteni zavislosti mame **sp\_depends** . Hodi se když chceme napr smazat tabulku. Zjistime tak jestli na tabulce není zavisla napr. nejaka procedura.

### Stored Procedures with output parameters

<https://www.youtube.com/watch?v=bldBshxuhMk>

Pouzivame klicove slovo output. Output parameter je vždy typu int.

Vytvoreni procedury s vystupem

CREATE PROC spGetTest2ValuesCount

@Value nvarchar (50),

@Count int output

AS

BEGIN

SELECT @Count = COUNT(Id) FROM dbo.Test2 WHERE Value = @Value

END

Pouziti

DECLARE @OutputParameter nvarchar(50)

EXEC spGetTest2ValuesCount 'YYYY' , @OutputParameter output

PRINT @OutputParameter

Změna procedury

ALTER PROC spGetTest2ValuesCount

@Value nvarchar (50),

@Count int output

AS

BEGIN

SELECT @Count = COUNT(Id) FROM dbo.Test2 WHERE Value = @Value

END

### Output parameters vs Return values

<https://www.youtube.com/watch?v=st8RnNg_LLA>

Když executneme proceduru vraci se nam Integer. Nula znaci succesfull operaci, jina hodnota nez nula znamena ze operace failed.

Output parameter

CREATE PROC spGetCountOfPersons

@TotalCount int output

AS

BEGIN

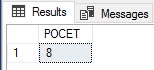
SELECT @TotalCount = COUNT(Id) from dbo.Person

END

DECLARE @Total int

EXEC spGetCountOfPersons @Total output

SELECT @Total AS POCET



Return value

CREATE PROCEDURE spGetCountOfPersonsReturnValue

AS

BEGIN

RETURN (SELECT COUNT(\*) FROM dbo.Person)

END

DECLARE @ReturnValue int

EXEC @ReturnValue = spGetCountOfPersonsReturnValue

PRINT @ReturnValue



Ren value je vždy typu int a nelze napsat storovku která vraci return value jineho typu. Napr nejde vratit jmeno podle Id:

ALTER PROC [dbo].[spSelectById]

@Id int

AS

BEGIN

return (SELECT Value FROM dbo.Test1 WHERE Id = @Id)

END

GO

DECLARE @Name nvarchar(50)

EXEC @Name = spSelectById 8

PRINT @Name

Místo toho musime pouzit output parameter

ALTER PROC [dbo].[spSelectById]

@Id int,

@Name nvarchar(50) output

AS

BEGIN

SELECT @Name = Value FROM dbo.Test1 WHERE Id = @Id

END

GO

DECLARE @Name nvarchar(50)

EXEC spSelectById 5 , @Name OUT

PRINT 'JMENO JE: '+ @Name

### While , If, Else, Set

Na SQL serveru se da decela dobře programovat :

DECLARE @Start int

SET @Start = 65

WHILE (@Start < 90)

BEGIN

PRINT CHAR(@Start)

SET @Start = @Start + 1

IF (@Start = 70)

PRINT ' Stop '

ELSE

PRINT @Start

END

### While iterace:

-- ABYCH MOHL V SQL MIT NECO JAKO List<int> POTREBUJU TABLE

-- DEKLARUJ VSE POTREBNE - TABLE KDE JE IDENTITA A JEDEN SLOUPEC (ID CKA KTERE POTREBUJU )

DECLARE @securables TABLE(idx INT PRIMARY KEY IDENTITY (1, 1), securableId int NOT NULL );

DECLARE @numrows INT, @i INT, @secId INT;

SET @i = 1

-- NAPLNIM TABULKU SELECTEM Z JINE TABULKY

INSERT INTO @securables

(

securableId

)

SELECT Id FROM Membership.Securable s WHERE s.Description LIKE 'SimLog%'

-- DO PROMENNE @numrows SI PRIRADIM POCET RADKU Z TABULKY

SET @numrows = (SELECT COUNT(\*) FROM @securables)

-- PROITERUJU

WHILE (@i <= @numrows) -- DOKUD @i JE MENSI NEZ POCET RADKU V TABULCE

BEGIN

SET @secId = (SELECT TOP(1) securableId FROM @securables WHERE idx = @i) -- V KAZDE ITERACI SETNI @secId

SELECT @secId -- VYPIS, PRIPADNE S NIM NECO PROVED

SET @i = @i + 1

END

### Pridani a odstraneni sloupce

ALTER TABLE PERSON ADD TestName nvarchar(100)

GO

ALTER TABLE Person DROP COLUMN TestName

### String Functions

<https://www.youtube.com/watch?v=qJFr-R76r9A>

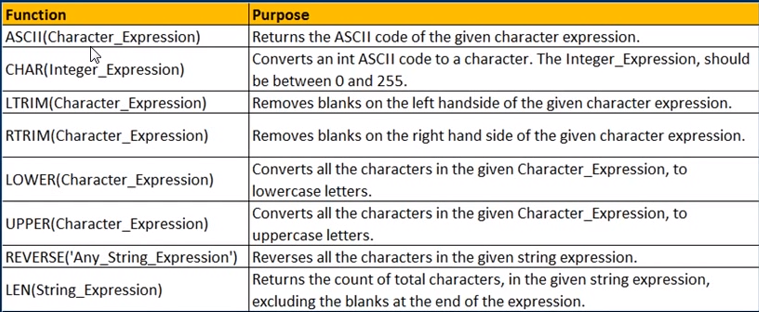
Nazev databaze -> Programmability -> Functions -> System Functions -> String Functions

V tomto umisteni jsou systemove funkce které muzeme vyuzivat napr nasledujici funkce vezme znak a vrati jeho int hodnotu . Když rozklikneme String Functions -> ASCII -> Parameters uvideme jake metoda potrebuje vstupy a vystup

SELECT ASCII('a') => prevede na int

select CHARINDEX('s', '12345s789') => najde index

select UPPER('ahoj') => prevede na velke pismena



Funkce je mozne do sebe i vnorovat

update Person set FullName = LTRIM(RTRIM(FirstName)) + ' '+LTRIM(RTRIM(LastName))

### LEFT, RIGHT, CHARINDEX and SUBSTRING functions in sql server Part 23

<https://www.youtube.com/watch?v=vN4sy5nHn6k>

### Insert do tabulky z deklarovaneho pole objektu

declare @listOfIDs table (id int);

insert @listOfIDs(id) values(1),(2),(3);

select \*

from TabA

where TabA.ID in (select id from @listOfIDs)

or

declare @listOfIDs varchar(1000);

SET @listOfIDs = ',1,2,3,'; --in this solution need put coma on begin and end

select \*

from TabA

where charindex(',' + CAST(TabA.ID as nvarchar(20)) + ',', @listOfIDs) > 0

Cenikovadlo:

CREATE TABLE ##Pricing\_TVMessageIds (Id INT NOT NULL, Ord INT NOT NULL IDENTITY (1,1))

declare @List table(Id int)

insert @List values (4),(5),(6);

insert into ##Pricing\_TVMessageIds (Id) select Id from @List

select \* from ##Pricing\_TVMessageIds

drop table ##Pricing\_TVMessageIds

viz:

[C:\Users\phlavenka\OneDrive\Nielsen prace\Moje poznamky Nielsen\SQL jak nasypat ListOfInt do db pomoci acessoru.docx](file:///C:\Users\phlavenka\OneDrive\Nielsen%20%20prace\Moje%20poznamky%20Nielsen\SQL%20jak%20nasypat%20ListOfInt%20do%20db%20pomoci%20acessoru.docx)

### CTE tabulky Common table expression

<https://www.essentialsql.com/introduction-common-table-expressions-ctes/>

A CTE (Common Table Expression) is temporary result set that you can reference within another SELECT, INSERT, UPDATE, or DELETE statement.

A CTE always returns a result set.

They are used to simplify queries, for example, you could use one to eliminate a derived table from the main query body

with ##Pricing\_TVMessageIdsCTE AS // Za nazev tabulky pridame CTE to je common table expression . Tabulku nikde nevytvarime ani nelikvidujeme

(

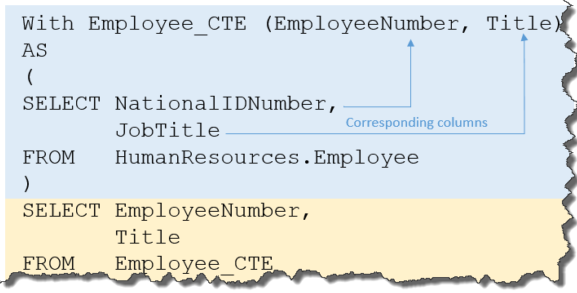
Select \*, ROW\_NUMBER () Over(Partition by Id order by Id) as RowNumber

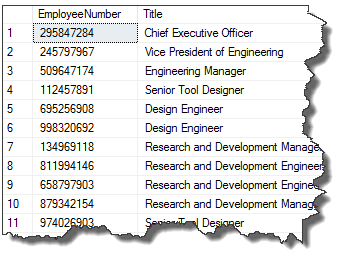
from ##Pricing\_TVMessageIds

)

Delete from ##Pricing\_TVMessageIdsCTE where RowNumber > 1

Muzu jeste za definici dat do zavorky jak se maji jmenovat sloupce CTE result setu:





### Jak odebrat duplicitni hodnoty z tabulky a nechat tam unikatni hodnoty

Chlalim Petre (vlastne oba).

Ad 2) – tento dotaz fungovat bude, jen si nejsem jistej ze zadani toho dotazu (neznam) bylo "smaz vsechny motivlety krome jednoho nahodneho motivletu za kazdeho ownera" ... neverim ze ti dal nekdo takove zadani :)

Ano, nahodneho – protoze kdyz neco Partitioninguju podle X a este navic pak orderuju (pro ucely ROW\_NUMBER()) pouze podle X, tak jsem vlastne rekl: pro kazde unikatni X mi nahodne serad zaznamy ...

Rikam to dobre PetoHol? :)

Doufam ze jste nedali za ukol Peterovi na to prijit sam a ja to ted cele proflaknul :)

CREATE TABLE ##Pricing\_TVMessageIds (Id INT NOT NULL, Ord INT NOT NULL IDENTITY (1,1))

declare @From datetime;

select @From = (select cast(AdvertisedFrom as date) from media.MediaMessage where Id = 9566145)

declare @To datetime;

select @To = (select DATEADD(DAY, 1, cast( AdvertisedTo as date)) from media.MediaMessage where Id = 9566145)

INSERT INTO ##Pricing\_TVMessageIds (Id) SELECT Id from media.MediaMessage where MediaTypeId = 2 and AdvertisedFrom >= @From and AdvertisedTo <= @To

//Tim ze insert udelam jeste jednou budou vsechny radky duplicitni

INSERT INTO ##Pricing\_TVMessageIds (Id) SELECT Id from media.MediaMessage where MediaTypeId = 2 and AdvertisedFrom >= @From and AdvertisedTo <= @To

; // Strednik nutny

with ##Pricing\_TVMessageIdsCTE AS // Za nazev tabulky pridame CTE to je common table expression . Tabulku nikde nevytvarime ani nelikvidujeme

(

Select \*, ROW\_NUMBER () Over(Partition by Id order by Id) as RowNumber

from ##Pricing\_TVMessageIds

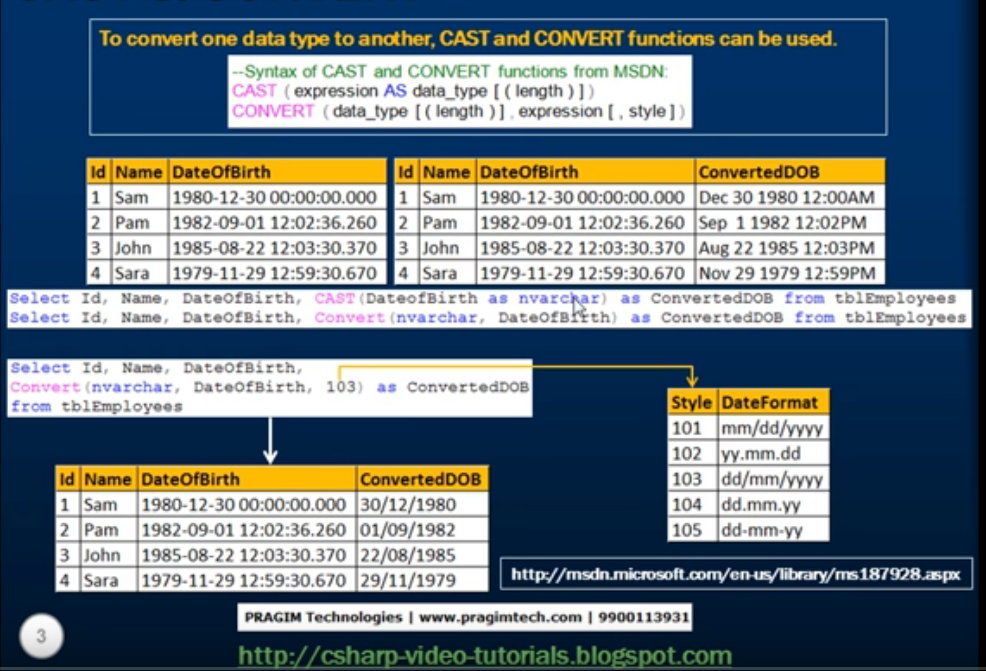
)

Delete from ##Pricing\_TVMessageIdsCTE where RowNumber > 1

SELECT COUNT(DISTINCT Id) FROM ##Pricing\_TVMessageIds

SELECT COUNT(Id) FROM ##Pricing\_TVMessageIds

### CAST & CONVERT



Cast (length) muzeme urcit delku vraceneho stringu :

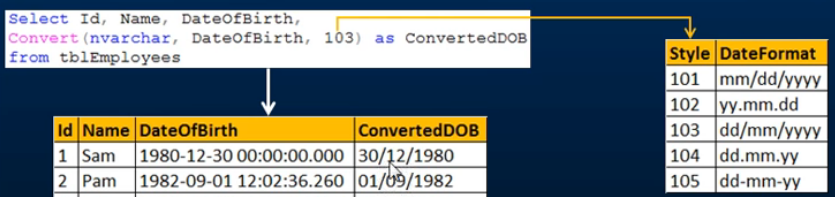








Convert (style) k dispozici jsou preddefinovane styly:



### CAST as bit

Cast muzeme pouzit na posuzovani bool hodnot

SELECT cast((1) \* (1) AS bit) => bude 1

SELECT cast((1) \* (0) AS bit) => bude 0

Diky teto funkci muzeme vytvorit bool hodnotu za dvou query, ktere si pomoci operatoru ISNULL vratime jako bool:

use MediaData3BGAuto

declare @mediumId tinyint, @dateTimeFrom datetime, @dateTimeTo datetime;

set @mediumId = '68'; --lze preskocit

--set @mediumId = '1'; --nelze preskocit

set @dateTimeFrom = '2019-04-12 01:00:00.000';

set @dateTimeTo = dateadd(hour, 1, @dateTimeFrom);

select CAST(

// první query

        ISNULL(

        (SELECT TOP 1 1

        FROM SimLog.BroadcastingDescription bd

        WHERE bd.MediumId IN (@mediumId) AND bd.DateTimeFrom > DATEADD(HOUR, -12, @dateTimeFrom) AND bd.DateTimeTo >= @dateTimeTo)

        , 0)

\* // hvezda pridava do porovnani další bool

        ISNULL(

// druha query

        (SELECT TOP 1 0

        FROM SimLog.IntervalRemark ir

        WHERE ir.MediumId IN (@mediumId) AND ir.[From] < @dateTimeTo AND ir.[To] > @dateTimeFrom)

        , 1)

// cely cast prekonvertime na bit

AS bit)

### FLOOR ()

Return the largest integer value that is equal to or less than a number:

Nejedna se o zaokrouhleni ROUND():

SELECT FLOOR(25.75) AS FloorValue; = 25

SELECT FLOOR(25.1) AS FloorValue; = 25

SELECT FLOOR(24.75) AS FloorValue; = 24

SELECT FLOOR(-13.5) AS FloorValue; = -14

Select (cast(getdate))

### DATEPART ()

Vraci cast datumu , navratovy typ je vzdy int

SELECT

DATEPART(YEAR, '2018-5-18 12:10:30.123'),

DATEPART(MONTH, '2018-5-18 12:10:30.123'),

DATEPART(DAY, '2018-5-18 12:10:30.123'),

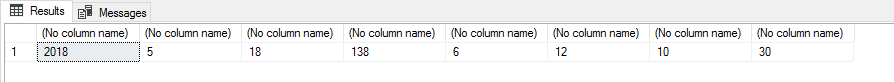
DATEPART(DAYOFYEAR, '2018-5-18 12:10:30.123'),

DATEPART(WEEKDAY, '2018-5-18 12:10:30.123'),

DATEPART(HOUR, '2018-5-18 12:10:30.123'),

DATEPART(MINUTE, '2018-5-18 12:10:30.123'),

DATEPART(SECOND, '2018-5-18 12:10:30.123')



Pouziti DATEPART() jako argument pro case (Holubec):

SELECT DISTINCT CASE WHEN DATEPART(HOUR, mm.AdvertisedFrom) < 6 THEN CAST(FLOOR(CAST(mm.AdvertisedFrom AS float)) AS DATETIME) ELSE CAST(FLOOR(CAST(DATEADD(DAY, 1, mm.AdvertisedFrom) AS float)) AS DATETIME) END,

DATEPART(DAY, mm.AdvertisedTo) , mm.MediumId

FROM Media.MediaMessage mm

WHERE mm.Id IN (93292651,

93292652,

93245844,

93292653)

### Jak dostat z DATETIME jen DATE

Holubec:

SELECT CAST(FLOOR(CAST(mm.AdvertisedFrom AS float)) AS DATETIME)

FROM Media.MediaMessage mm

WHERE mm.Id IN (93292651,

93292652,

93245844,

93292653)

1. Nejprve precastime datum na float:

CAST(mm.AdvertisedFrom as float)

1. Odstranime casovou cast datumu tak, ze z floatu udelame nejmensi mozny int viz FLOOR () :

FLOOR(CAST(mm.AdvertisedFrom as float))

1. Precastime zpet na DateTime:

CAST(FLOOR(CAST(mm.AdvertisedFrom as float)) AS datetime)

### Ziskani jen datumu z datetime prevedenim na float, pomoci floor()a cast()

* Mame datum

select GETDATE()



* Datetime si prevedeme na float

select cast(getdate() as float)



* Mame desetinne cislo. Cela cast predstavuje datum a cast za carkou predstavuje hodiny. Hodin se chceme zbavit, takze udelame z desetinneho cisla cislo cele.

select floor(cast(GETDATE() as float))



* Cele cislo zase prevedeme zpet na dateTime:

select cast(floor(cast(getdate() as float)) as datetime)



* Pokud nechci dnesni den ale nasledujici pouziju ceiling, ktery mi da cele cislo nahoru od desetinneho (floor dava dolni cele cislo)

select cast(ceiling(cast(getdate() as float)) as datetime)



### Ceiling()

Je opakem Floor()

### Coalesce()

<https://www.w3schools.com/sql/func_sqlserver_coalesce.asp>

The COALESCE() function **returns** the **first** **non-null expression** in a list.



### Scalar user defined FUNCTIONs part 30

<https://www.youtube.com/watch?v=OV5CquR1Svo&t=208s>

Rozdil mezi funkci a storovanou procedurou je, ze **storovka** **nemuze** byt pouzita v select a where class.

Muzeme vytvorit Function ktera pocita vek na zaklade zadaneho datumu :

create function CalculateAge (@DOB Date)

returns int

as

begin

declare @Age int

set @Age =

DATEDIFF(year, @DOB, getdate())-

case

when (Month (@DOB) > Month(GETDATE())) or

(Month (@DOB) = Month(GETDATE()) and day (@DOB) > day (GETDATE()))

then 1

else 0

end

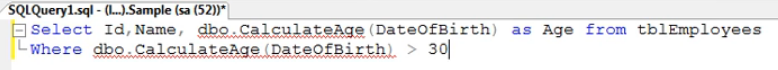
return @Age

end

Pouziti:

select MediaData3Auto.dbo.CalculateAge('1999-03-23')

Uplne stejne by se dala vytvorit storovka, ale **nedala** by se pouzit v subSelectu:

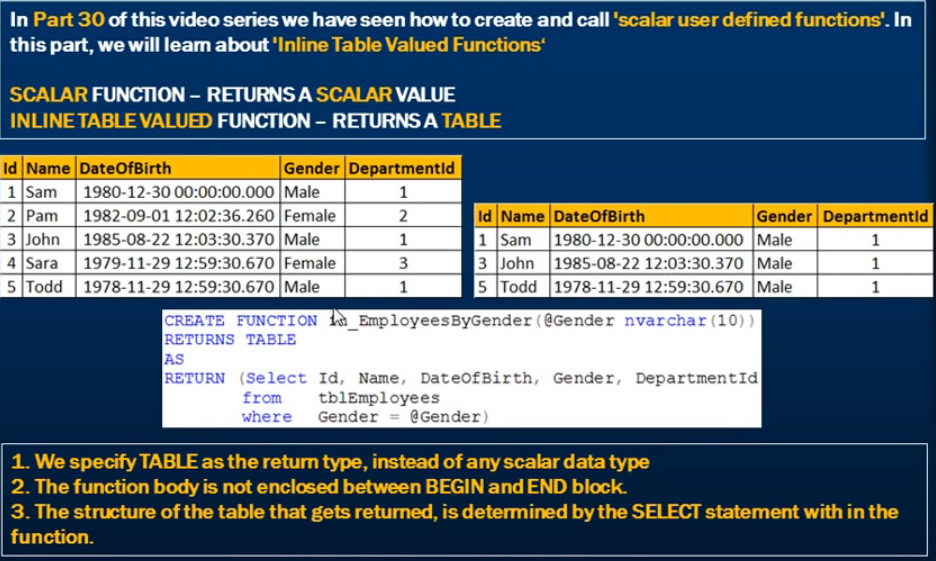


To ze je to podtrzene nijak nevadi, jen se musi zadat nazev databaze a nazev funkce. Muze tam byt mezi dbo.

### Inline table valued functions in sql server Part 31

<https://www.youtube.com/watch?v=hs4mReAzESc>





Create table Employees (Nam nvarchar(10), Birdth datetime, Gender nvarchar(10), depNumber int ) //vytvoreni test tabulky

Insert into Employees values ('Petr', '1980-03-23', 'Male', 7)

Insert into Employees values ('Renata', '2006-06-16', 'Female', 6)

Insert into Employees values ('Tana', '2009-09-16', 'Female', 5)

Select \* from Employees

Create function EmployeesByGender(@Gender nvarchar(10)) // vytvoreni funkce

returns table // vraci table

as

return (Select Nam, Birdth, Gender, depNumber

from Employees

where Gender = @Gender)

select \* from MediaData3Auto.dbo.EmployeesByGender('Female') // volani funkce

drop table Employees

// nezapomenout smazat vytvorene funkce

### Scope identity scope\_identity() vs @@IDENTITY

Prvni priklad:

USE ##db;

GO

-- **Vytvoreni tabulky A**

CREATE TABLE A (

A\_id int IDENTITY(1,1)PRIMARY KEY,

A\_name varchar(20) NOT NULL);

INSERT A

VALUES ('Lisa'),('Mike'),('Carla');

SELECT \* FROM A;

-- **Vytvoreni tabulky B**

CREATE TABLE B (

B\_id int IDENTITY(100,5)PRIMARY KEY, -- zacina od 100, pridava po peti

B\_name varchar(20) NULL);

INSERT B (B\_name)

VALUES ('boathouse'), ('rocks'), ('elevator');

SELECT \* FROM B;

**-- Trigger - spustit samostatne--**

Create trigger ATrig

on A

for insert as

Begin

insert B values ('')

end;

-- Insert do tabulky A --

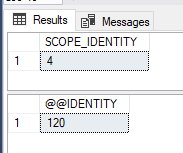
INSERT A VALUES ('Rosalie');

SELECT SCOPE\_IDENTITY() AS [SCOPE\_IDENTITY];

GO

SELECT @@IDENTITY AS [@@IDENTITY];

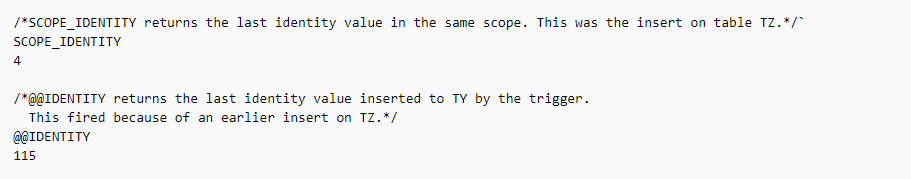
GO



**Pokud je na tabulce trigger ktery insertuje do jine tabulky, vrati mi @@IDENTITY hodnotu identity z tabulky B**

**SCOPE\_IDENTITY mi vrati hodnotu identity pro tabulku A**

<https://docs.microsoft.com/en-us/sql/t-sql/functions/scope-identity-transact-sql?view=sql-server-2017>



Druhy priklad:

USE tempdb

create table Person

(

Id int IDENTITY not null Primary key,

Gender nvarchar(50) null

)

create table tblGender

(

Id int IDENTITY not null Primary key,

Gender int null

)

INSERT INTO dbo.Person(Gender)VALUES('Male')

INSERT INTO dbo.Person(Gender)VALUES('FeMale')

SELECT scope\_identity() AS ScopeIdentityPoDruhemInsertuDoTabulkyPerson

INSERT INTO dbo.tblGender(Gender)VALUES(100)

SELECT scope\_identity() AS ScopeIdentityInsertuDoTabulkyGender

--SELECT \* FROM dbo.Person p

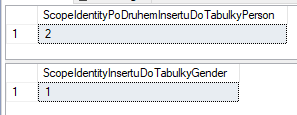
--SELECT \* FROM tblGender

USE tempdb

DROP TABLE Person

DROP TABLE dbo.tblGender

SELECT 'Scope identity vrati ID posledniho insertu bez zavislosti na to, kolikrat bylo predtim insertnuto do jine tabulky'



### Zaokrouhleni datetime na milisekundy pomoci CASE

declare @date datetime;

select @date = '2018-06-27 16:18:30.498'

select (

case

when datepart(ms, @date) >= 500

then dateadd(ms, 1000-datepart(ms, @date), @date)

else dateadd(ms, -datepart(ms, @date), @date)

end

)

Pozor je tu jedna zrada SQL Server moc neumi milisekundy takze .499 zaokrouhli nahoru a .498 dolu



### Nepovinny parametr ve storovce v @Sql na zaklade if

Ve schvalovadle jsem upravoval storovku proc\_AutoImporter\_ListMediumCodingInfo

Cela storovka zde

if exists(select \*

from information\_schema.routines

where specific\_schema = N'Media'

and specific\_name = N'proc\_AutoImporter\_ListMediumCodingInfo'

)

drop procedure [Media].[proc\_AutoImporter\_ListMediumCodingInfo];

GO

/\*

\* Účelem procedury je zjistit stav okódování médíí vybraných na základě zadaných kritérií.

\*/

create procedure [Media].[proc\_AutoImporter\_ListMediumCodingInfo]

(

@StartDate datetime,

@EndDate datetime,

@MediaTypeId smallint = null,

@MediumId smallint = null

)

as

begin

declare @sql nvarchar(3000),

@params nvarchar(500),

@dailyPressPeriodicityId int;

select @dailyPressPeriodicityId = 2;

select @sql = N'select MediumId, MediumName, MediaTypeId, MediaTypeName, AdvertisedDate,

case when PressPeriodicityId = @dailyPressPeriodicityId then 1 else 0 end as IsDailyPressPeriodicity,

coalesce([0], 0) as X, coalesce([1], 0) as O, coalesce([2], 0) as C,

coalesce([4], 0) as B, coalesce([8], 0) as A, coalesce([16], 0) as D, AdvertisementSource

from (

select mm.MediumId, min(mv.Name) as MediumName,

(case when (@MediaTypeId=2 and datepart(hh, mm.AdvertisedFrom) < 6) then dateadd(day,-1, convert(datetime, floor(convert(float, mm.AdvertisedFrom)))) else convert(datetime, floor(convert(float, mm.AdvertisedFrom))) end) as AdvertisedDate,

mm.MediaTypeId, min(mt.Name) as MediaTypeName, mm.CodingPlausibilityId, count(CodingPlausibilityId) as PlausibilityCount, min(mv.PressPeriodicityId) as PressPeriodicityId, min(mm.AdvertisementSourceId) as AdvertisementSource

from Media.MediaMessage mm

join Media.Medium m on mm.MediumId = m.Id

join Media.MediaType mt on mm.MediaTypeId = mt.Id

join Media.MediumVersion mv on m.Id = mv.MediumId and mv.ActiveFrom <= mm.AdvertisedFrom and mv.ActiveTo > mm.AdvertisedFrom

where

(case when datepart(ms, mm.AdvertisedFrom) >= 500 then dateadd(ms, 1000-datepart(ms, mm.AdvertisedFrom), mm.AdvertisedFrom) else dateadd(ms, -datepart(ms, mm.AdvertisedFrom), mm.AdvertisedFrom) end) >= @StartDate

and (case when datepart(ms, mm.AdvertisedFrom) >= 500 then dateadd(ms, 1000-datepart(ms, mm.AdvertisedFrom), mm.AdvertisedFrom) else dateadd(ms, -datepart(ms, mm.AdvertisedFrom), mm.AdvertisedFrom) end) < @EndDate

and Ready = 0';

if @MediaTypeId is not null

select @sql = @sql + N' and mm.MediaTypeId = ' + cast(@MediaTypeId as nvarchar);

if @MediumId is not null

select @sql = @sql + N' and mm.MediumId = ''' + cast(@MediumId as char(36)) + '''' ;

select @sql = @sql + N' group by (case when (@MediaTypeId=2 and datepart(hh, mm.AdvertisedFrom) < 6) then dateadd(day,-1, convert(datetime, floor(convert(float, mm.AdvertisedFrom)))) else convert(datetime, floor(convert(float, mm.AdvertisedFrom))) end), mm.MediaTypeId, mm.MediumId, mm.CodingPlausibilityId) t';

select @sql = @sql + N' pivot (sum(PlausibilityCount) for CodingPlausibilityId in ([0], [1], [2], [4], [8], [16], [32])) as pt

order by MediaTypeId, MediumName, AdvertisedDate';

--print @sql;

--print len(@sql);

select @params = N'@StartDate datetime,@EndDate datetime,@dailyPressPeriodicityId int,@MediaTypeId int';

exec sp\_executesql @sql, @params,

@StartDate = @StartDate,

@EndDate = @EndDate,

@dailyPressPeriodicityId = @dailyPressPeriodicityId,

@MediaTypeId = @MediaTypeId;

end;

GO

grant exec on [Media].[proc\_AutoImporter\_ListMediumCodingInfo] to MediaDataBasicAccess;

-- [Media].[proc\_AutoImporter\_ListMediumCodingInfo] @StartDate = '2011-07-10 06:00:00', @EndDate = '2011-07-11 06:00:00', @MediaTypeId = 2, @MediumId = null;

Muzu si definovat ve storovce sql query ktere ma where , ke ktere muzu pripojit dodatek na zaklade parametru.

if exists(select \*

from information\_schema.routines

where specific\_schema = N'Media'

and specific\_name = N'proc\_test'

)

drop procedure [Media].[proc\_test]

GO

create procedure Media.proc\_test

as

begin

declare @sql nvarchar (3000);

declare @maximum int;

select @maximum = 80 Pokud sem dam vetsi cislo nez je podminka, @sql se zmeni

select @sql = N'select top (100)\* from media.Motivlet where Id > 300';

if @maximum > 100

select @sql = @sql + N' and Id < ' + cast (@maximum as nvarchar);

select @sql

execute (@sql) vypise tabulku 

end

exec Media.proc\_test

Pro @maximum < 100:



Pro @maximum >= 100:



### Jak dropnout uz existujici storovku

Pokud pisu storovku, nad ni dam nasledujici kod. Ten zajisti dropnuti storovky pokazde kdyz se upravi script a pusti. Stara se smaze a nova procedura vytvori:

if exists(select \*

from information\_schema.routines

where specific\_schema = N'Media'

and specific\_name = N'proc\_AutoImporter\_ListMediumCodingInfo'

)

drop procedure [Media].[proc\_AutoImporter\_ListMediumCodingInfo];

GO

### Jak executnout storovku ktera ma vstupni parametry

Pokud narazim na storovku ktera ma vstupni parametry

exec [Media].[proc\_AutoImporter\_ListMediumCodingInfo] @StartDate = '2017 -06-12', @EndDate = '2018 -06-14', @MediaTypeId = 2, @MediumId = 25



### Grant

pro uvedeného uživatele (nebo skupinu), povolí nějakou operaci (např. select) nad uvedenou tabulkou.

Viz

[C:\Users\phlavenka\OneDrive\Nielsen prace\Moje poznamky Nielsen\Vyjimky Exceptions a jejich reseni.docx](file:///C:\Users\phlavenka\OneDrive\Nielsen%20%20prace\Moje%20poznamky%20Nielsen\Vyjimky%20Exceptions%20a%20jejich%20reseni.docx)

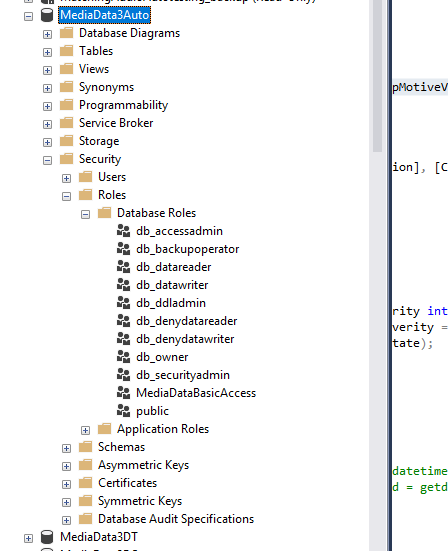
**BLToolkit.Data.DataException:** 'The SELECT permission was denied on the object 'TvImportItem', database 'MediaData3Auto', schema 'Import'.

GRANT SELECT, INSERT, UPDATE, DELETE TO MediaDataBasicAccess

Pokud pomoci scriptu tvorim novou tabulku v databazi, patri pod ni grant

grant select, insert, update, delete on Media.TempMotiveVersion to MediaDataBasicAccess;

### Db Role Database Roles



[‎6/‎28/‎2018 4:31 PM] Petr Dobeš:

Ahoj, to je chybějící grant nad db. Spusť v mangement studiu jednorázově:

GRANT SELECT, INSERT, UPDATE, DELETE TO MediaDataBasicAccess

a mělo by fungovat

[‎6/‎28/‎2018 4:31 PM] Peter Hlavenka:

dekuju

[‎6/‎28/‎2018 4:32 PM] Petr Dobeš:

počkej...

GRANT SELECT, INSERT, UPDATE, DELETE ON Import.TvImportItem TO MediaDataBasicAccess

ted je to správně

[‎6/‎28/‎2018 4:33 PM] Peter Hlavenka:

funguje i ten prvni :)  byl jsem rychlejsi :)

dik

[‎6/‎28/‎2018 4:33 PM] Petr Dobeš:

nz

[‎6/‎28/‎2018 4:34 PM] Peter Hlavenka:

vysvetlil bys mi v rychlosti co ten grant dela ?

[‎6/‎28/‎2018 4:37 PM] Petr Dobeš:

pro uvedeného uživatele (nebo skupinu), v tomto případě mediaDataBasicAccess, povolí nějakou operaci (např. select) nad uvedenou tabulkou.

..ale divný že zmizel. To je většinou potřeba jen u nových tabulek

[‎6/‎28/‎2018 4:38 PM] Peter Hlavenka:

mediaDataBasicAccess  je definovane kde ?

[‎6/‎28/‎2018 4:39 PM] Peter Hlavenka:

a jak jsi vedel ze chybi prave mediaDataBasicAccess ?

[‎6/‎28/‎2018 4:42 PM] Petr Dobeš:

Koukni v man. studiu v object exploreru: MediaData3Auto - Security - Roles - Database Roles. Je to db role která uskupuje některé uživatele v MediaData3Auto - Security - Users. Tys potřeboval přístup pro MediaDataNormingUser a ten je součástí.

součástí MediaDataBasicAcces

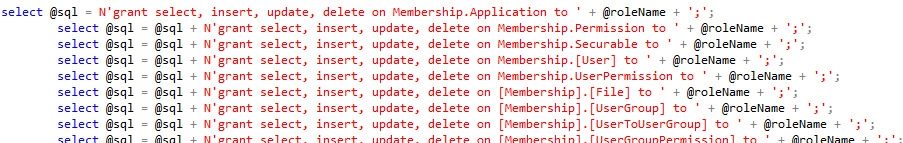
grant jde přidat i pro konrétní uživatele, ale takhle to mají rovnou všichni

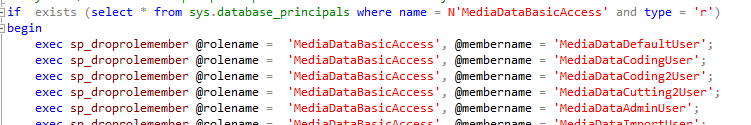
[‎6/‎28/‎2018 4:43 PM] Peter Hlavenka:

moc dekuju

[‎6/‎28/‎2018 4:48 PM] Petr Dobeš:

Kdyžtak pro zajímavost v repozitáří ...data\DatabaseModel\MediaData.Database\Create Scripts\Installation v těch .sql skriptech to je. Např. "**007 MD.Grants.sql**" jsou ty granty, "**900 MD.Users.sql**" tam je to vytváření uživatelů a rolí





### Row\_Number()

<https://www.youtube.com/watch?v=cvrwOoGwgz8>

Seskupi data v tabulce do serazenych skupin a ocisluje radky

Syntaxe :

V kazdem row\_number() musim mit Order by protoze rowNumber pracuje nad serazenymi daty.

Serazeni bez rozdeleni do casti:

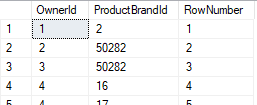
Pokud v over casti necham jen OrderBy, dostanu ve sloupci RowNumber jen serazene radky od 1.

SELECT TOP(100)

OwnerId,

ProductBrandId ,

ROW\_NUMBER() OVER (ORDER BY OwnerId) AS RowNumberFROM Media.Motivlet



Rozdeleni do skupin – Partition

Kdyz do over klauzule pridam PARTITION BY rozdeli se mi vysledek do serazenych skupin. RowNumber se resetne na 1 s kazdou novou skupinou.

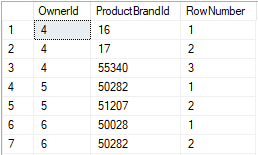
SELECT TOP(100)

OwnerId,

ProductBrandId ,

ROW\_NUMBER() OVER (PARTITION BY OwnerId ORDER BY OwnerId) AS RowNumber

FROM Media.Motivlet



Pouziti

Daji se tim odstranit duplicity z tabulek

Kdybych chtel z tabulky motivlet odstranit motivlety, ktere maji duplicitniho ownera rekl bych:

WITH Media.MotivletCTE as

(

SELECT

OwnerId,

ProductBrandId ,

ROW\_NUMBER() OVER (PARTITION BY OwnerId ORDER BY OwnerId) AS RowNumber

FROM Media.Motivlet

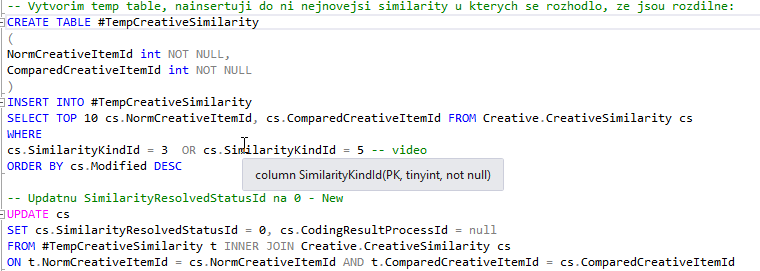
)

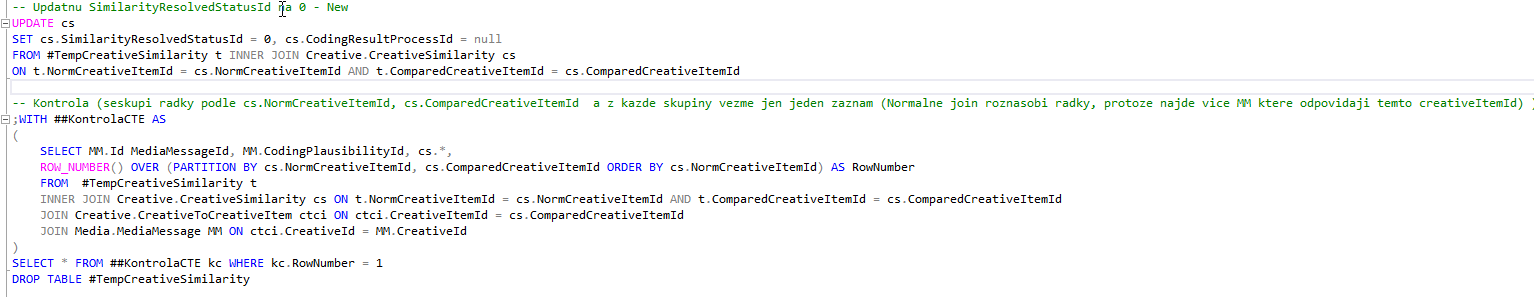
DELETE FROM Media.MotivletCTE WHERE RowNumber > 1

Pro kazdeho ownera by v databazi zustal jen jeden motivlet (podle toho jake zvolim razeni – napr ORDER BY OwnerId, CompanyBrandId)

**Pouziti napr na seskupeni radku podle CreativeSimilarity – od kazde skupiny chceme jen jeden radek dle partition by:**

**C:\Users\phlavenka\OneDrive\Nielsen prace\Moje poznamky Nielsen\My Scripts\Kodovadlo\Creative.Similarity\Nalezeni a update similarity podle MM.sql:**





Vymazani duplicit z media.SelfPromotionDefinition

;

with ##SelfPromotionDefinitionCTE AS

(

Select \*, ROW\_NUMBER () Over(Partition by Media.SelfPromotionDefinition.OwnerId, Media.SelfPromotionDefinition.MediumId, Media.SelfPromotionDefinition.ActiveFrom, Media.SelfPromotionDefinition.ActiveTo order by Media.SelfPromotionDefinition.Created) as RowNumber

from [Media].[SelfPromotionDefinition]

)

SELECT \* FROM ##SelfPromotionDefinitionCTE WHERE ##SelfPromotionDefinitionCTE.RowNumber > 1

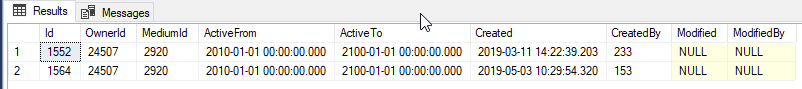
--Delete from ##SelfPromotionDefinitionCTE where RowNumber > 1

**Vrati tohle:**



**Vezmu prvni radek:**

SELECT \* FROM [Media].[SelfPromotionDefinition] spd WHERE spd.OwnerId = 24507 AND spd.MediumId = 2920

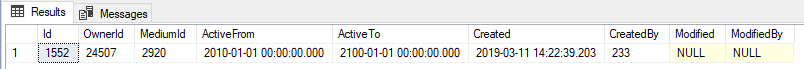


**Chci nechat jen jeden, a to ten starsi, tj to mensi Id (1552). Zkusim vymazat jen tento jeden radek:**

Delete from ##SelfPromotionDefinitionCTE where RowNumber > 1 and MediumId = 2920

**Vysledek:**

SELECT \* FROM [Media].[SelfPromotionDefinition] spd WHERE spd.OwnerId = 24507 AND spd.MediumId = 2920



**Works! Muzu smazat vsechny.**

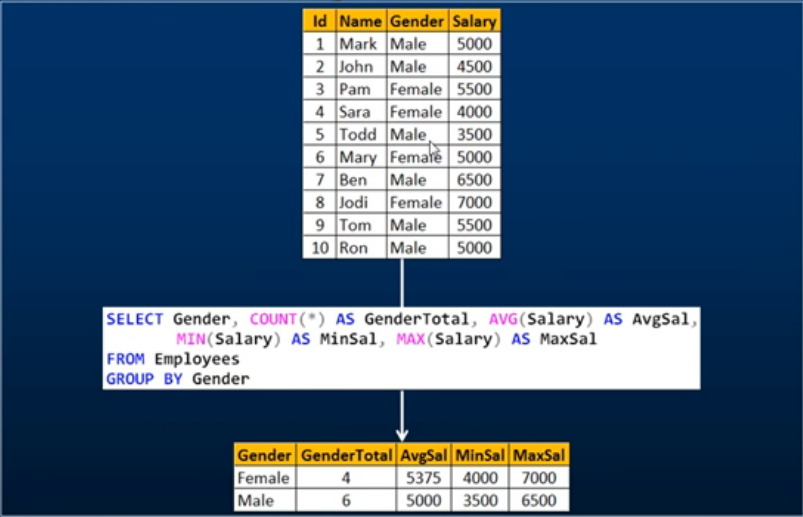
### Over clause

<https://www.youtube.com/watch?v=KwEjkpFltjc>

Kdyz chci z tabulky nejake agregovane data ale vypsane v kazdem radku tabulky pouziju over.

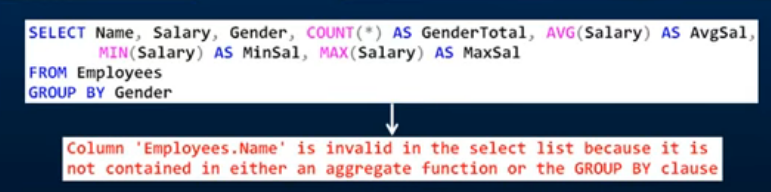
Tohle resi problem GroupBy, kde muzu dostat pro nejakou kategorii pocet radku, ktere splnuji group klauzuli

Priklad : Pomoci group by dostaneme prumernou mzdu pro kazde pohlavi

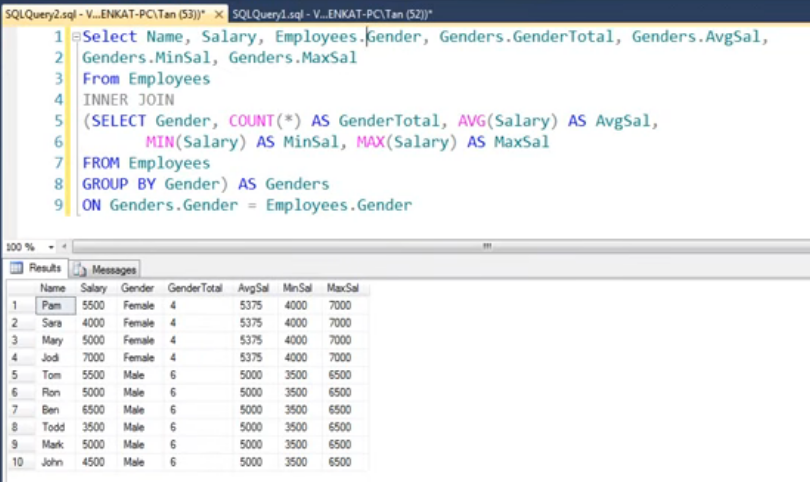


Co kdybych chtel tyto pohlavi vypsat radek po radku a v kazdem bych chtel videt prumernou mzdu pro dane pohlavi?

Kdyz doplnim group by dostanu chybu, ze sloupec neni zahrnuty do group by clause.



Muzu z group by udelat subQuery a najoinovat ji na jiny select



Nebo pouzit klauzuli Over :

SELECT TOP(100)

OwnerId,

ProductBrandId ,

CompanyBrandId,

Count(CompanyBrandId) over(PARTITION BY OwnerId) AS CountOfCompanyBrands

FROM Media.Motivlet m



### RANK() DENSERANK()

<https://www.youtube.com/watch?v=5-La_uSNkKU>

rank

ČEŠTINA

hodnost

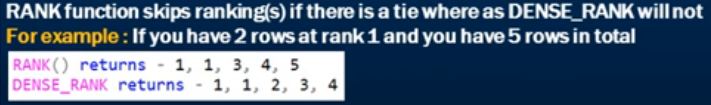
|  |  |
| --- | --- |
| podstatné jméno | pořadí, hodnost, postavení, řada, stupeň, řád, linie |

Syntaxe :



Je to hodne podobne Row\_Number() s tim rozdilem, ze kdyz pouziju rank a mam stejne hodnoty ve sloupci tak pro radky se stejnou hodnotou dostanu stejne cislo.

Rozdil mezi rank a dense je ten, ze rank nechava diry podle toho kolik shodnych radku najde. Zde chybi pro rank dvojka: (dva radky byly stejne takze rank pocita jedna, dva, a na treti pozici dosadi trojku.) (dense pocita jedna, jedna pro stejne radky a na treti pozici da dvojku)



Pokud navic pouziju Partition By, rank a dense\_rank se mi resetne na 1 s kazdou novou skupinou.

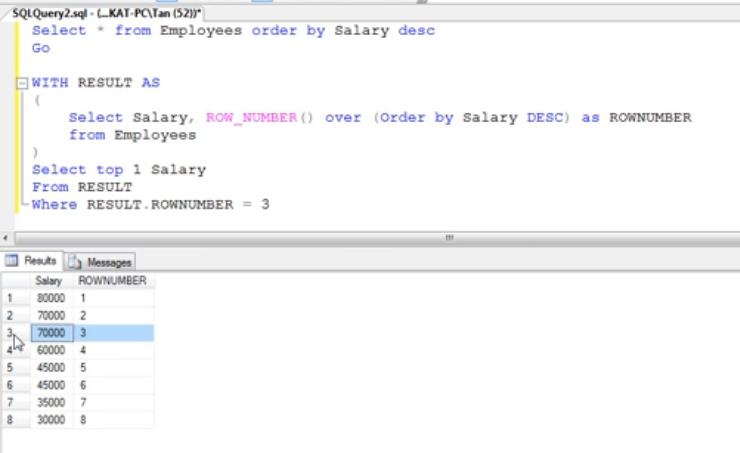
Pouziti:

Pomoci ranku muzeme zjistit nejvyssi hodnotu nektereho sloupce v ramci skupiny:

### WITH Result AS

<https://www.youtube.com/watch?v=fvPddKyHxpQ>

Dostane treti nejvyssi plat z tabulky Employees



### Insert from one table to another table (Insert Into Select)

CREATE TABLE ##MyTempTable

(

Id int NOT NULL IDENTITY(1,1),

MediumId int NOT NULL ,

PriceValue money NOT NULL,

OriginalId nvarchar(1000)

)

INSERT INTO ##MyTempTable (MediumId, PriceValue, OriginalId)

SELECT TOP 100

mm.MediumId,

mm.PriceValue,

m.OriginalId

FROM Media.MediaMessage mm

JOIN Media.Medium m ON mm.MediumId = m.Id

WHERE mm.ImportId IS NOT NULL AND m.OriginalId LIKE '%A%'

SELECT \* FROM ##MyTempTable

DROP ##MyTempTable

### Jak zjistit Ip adresu serveru

SELECT @@SERVERNAME, dec.local\_net\_address

FROM sys.dm\_exec\_connections AS dec

WHERE dec.session\_id = @@SPID;

### Snapshot a reset databaze

-- create snapshot MEDIADATA

USE master

CREATE DATABASE MediaData3Auto\_ss ON

( NAME = N'MediaData', FILENAME = N'd:\SNAPSHOTS\MediaData3Auto\_ss' ),

( NAME = N'DefaultCreativeData', FILENAME = N'd:\SNAPSHOTS\MediaData3Auto\_ss\_DefaultCreativeData' ),

( NAME = N'VideoCreativeData', FILENAME = N'd:\SNAPSHOTS\MediaData3Auto\_ss\_VideoCreativeData' ),

( NAME = N'VideoCreativeData\_2', FILENAME = N'd:\SNAPSHOTS\MediaData3Auto\_ss\_VideoCreativeData25' ),

( NAME = N'ftrow\_ft', FILENAME = N'd:\SNAPSHOTS\MediaData3Auto\_ss\_ftrow\_ft' )

AS SNAPSHOT OF MediaData3Auto;

GO

-- restore to snapshot

USE master

EXEC dbo.killusers 'MediaData3Auto'

EXEC dbo.killusers 'MediaData3Auto\_ss'

RESTORE DATABASE MediaData3Auto from

DATABASE\_SNAPSHOT = 'MediaData3Auto\_ss';

--Skript na DT db by měl být jen: (tam je snapshotem vlastně RO db):

USE master

EXEC dbo.killusers 'MediaData3DT'

RESTORE DATABASE MediaData3DT from

DATABASE\_SNAPSHOT = 'MediaData3RO';

GO

-- na Auto je potreba si na zacatku tydne vytvorit snapshot, ke kteremu se da vracet

-- na DT je snapshot a staci restornout

Hobbys vytvoril storovku ktera nas od tohoto odstini. Jako parametr predame nazev databaze kterou chceme restorovat.

**exec master.[dbo].[RestoreDBFromSnapshot] @DB\_NAME = 'MediaDataRO'**

U CZ RC musime nejprve pustit storovku :

* EXEC RestoreDBFromSnapshot 'MediaData3RC'

A pak na ostrej db (Alfred) pustit tuto query (jinak nepujde videomatching atd.. ) :

EXEC('

use MediaData3RC

update [dbo].[Params] set Value = ''itservices@mediaresearch.cz'' where value like ''%@%''

update [dbo].[Params] set Value = ''net.tcp://192.168.0.122:8001/VideoMatchingSortingService'' where name = ''PREPROCESSING\_ENDPOINTS''

update [Environment].[ServiceComponentInstance] set [Value02] = ''stoupa'', [Value03]=''MediaData3RC'' where id in (8,9)

update [Environment].[ServiceComponentInstance] set [Value01] = 255 where id = 4

DROP SYNONYM [Source].[wanAgregationTargetGroup]

DROP SYNONYM [Source].[wanRule]

DROP SYNONYM [Source].[wanRuleSet]

DROP SYNONYM [Synonym].[Part]

DROP SYNONYM [Synonym].[Page]

drop synonym [Synonym].[Location]

create SYNONYM [Synonym].[Part] FOR [PrintStorageAuto].[dbo].[Part]

create SYNONYM [Synonym].[Page] FOR [PrintStorageAuto].[dbo].[Page]

CREATE SYNONYM [Synonym].[Location] FOR [Stoupa].[TvStorage2Auto].[dbo].[Location];

create synonym [Source].[wanAgregationTargetGroup] for [Stoupa].[PoolingAuto].[SimProducer].[wanAgregationTargetGroup]

create synonym [Source].[wanRule] for [Stoupa].[PoolingAuto].[SimProducer].[wanRule]

create synonym [Source].[wanRuleSet] for [Stoupa].[PoolingAuto].[SimProducer].[wanRuleSet]

grant select, insert, update, delete on [Synonym].[Location] to MediaDataBasicAccess;

grant select, insert, update, delete on [Synonym].[Page] to MediaDataBasicAccess;

grant select, insert, update, delete on [Synonym].[Part] to MediaDataBasicAccess;

grant select, insert, update, delete on [Source].[wanAgregationTargetGroup] to MediaDataBasicAccess;

grant select, insert, update, delete on [Source].[wanRule] to MediaDataBasicAccess;

grant select, insert, update, delete on [Source].[wanRuleSet] to MediaDataBasicAccess

if exists (select \* from sys.database\_principals where name = N''MediaDataThumbnailUser'')

drop user [MediaDataThumbnailUser];

create user [MediaDataThumbnailUser] for login [MediaDataThumbnailUser] with default\_schema = [Media];

exec sp\_addrolemember @rolename = N''MediaDataBasicAccess'', @membername = N''MediaDataThumbnailUser''

if exists (select \* from sys.database\_principals where name = N''MediaDataPricingUser'')

drop user MediaDataPricingUser;

create user MediaDataPricingUser for login MediaDataPricingUser with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataPricingUser''

if exists (select \* from sys.database\_principals where name = N''MediaDataChangingUser'')

drop user MediaDataChangingUser;

create user MediaDataChangingUser for login MediaDataChangingUser with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataChangingUser''

if exists (select \* from sys.database\_principals where name = N''MediaDataExportCreativeUser'')

drop user [MediaDataExportCreativeUser];

create user [MediaDataExportCreativeUser] for login [MediaDataExportCreativeUser] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataExportCreativeUser''

if exists (select \* from sys.database\_principals where name = N''MediaDataCuttingUser'')

drop user [MediaDataCuttingUser];

create user [MediaDataCuttingUser] for login [MediaDataCuttingUser] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataCuttingUser''

if exists (select \* from sys.database\_principals where name = N''MediaDataCutting2User'')

drop user [MediaDataCutting2User];

create user [MediaDataCutting2User] for login [MediaDataCutting2User] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataCutting2User''

if exists (select \* from sys.database\_principals where name = N''PrintStorageScanningUser'')

drop user [PrintStorageScanningUser];

create user [PrintStorageScanningUser] for login [PrintStorageScanningUser] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''PrintStorageScanningUser''

if exists (select \* from sys.database\_principals where name = N''MediaDataAdminUser'')

drop user [MediaDataAdminUser];

create user [MediaDataAdminUser] for login [MediaDataAdminUser] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataAdminUser''

if exists (select \* from sys.database\_principals where name = N''MediaDataApprovingUser'')

drop user [MediaDataApprovingUser];

create user [MediaDataApprovingUser] for login [MediaDataApprovingUser] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataApprovingUser''

if exists (select \* from sys.database\_principals where name = N''MediaDataCatchingUser'')

drop user [MediaDataCatchingUser];

create user [MediaDataCatchingUser] for login [MediaDataCatchingUser] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataCatchingUser''

if exists (select \* from sys.database\_principals where name = N''MediaDataCoding2User'')

drop user [MediaDataCoding2User];

create user [MediaDataCoding2User] for login [MediaDataCoding2User] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataCoding2User''

if exists (select \* from sys.database\_principals where name = N''MediaDataNormingUser'')

drop user [MediaDataNormingUser];

create user [MediaDataNormingUser] for login [MediaDataNormingUser] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataNormingUser''

if exists (select \* from sys.database\_principals where name = N''MediaDataScheduleWinUser'')

drop user [MediaDataScheduleWinUser];

create user [MediaDataScheduleWinUser] for login [MediaDataScheduleWinUser] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataScheduleWinUser''

if exists (select \* from sys.database\_principals where name = N''MediaDataImportUser'')

drop user [MediaDataImportUser];

create user [MediaDataImportUser] for login [MediaDataImportUser] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataImportUser''

if exists (select \* from sys.database\_principals where name = N''MediaDataImportingUser'')

drop user [MediaDataImportingUser];

create user [MediaDataImportingUser] for login [MediaDataImportingUser] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataImportingUser''

if exists (select \* from sys.database\_principals where name = N''MediaDataCoding2User'')

drop user [MediaDataCoding2User];

create user [MediaDataCoding2User] for login [MediaDataCoding2User] with default\_schema = [dbo];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataCoding2User''

if exists (select \* from sys.database\_principals where name = N''SimLogUser'')

drop user [SimLogUser];

create user [SimLogUser] for login [SimLogUser] with default\_schema = [dbo];

exec sp\_addrolemember @rolename = N''MediaDataBasicAccess'', @membername = N''SimLogUser'';

if exists (select \* from sys.database\_principals where name = N''MediaDataReprocessingUser'')

drop user [MediaDataReprocessingUser];

create user [MediaDataReprocessingUser] for login [MediaDataReprocessingUser] with default\_schema = [dbo];

exec sp\_addrolemember @rolename = N''MediaDataBasicAccess'', @membername = N''MediaDataReprocessingUser'';

if exists (select \* from sys.database\_principals where name = N''MediaDataMatching1User'')

drop user [MediaDataMatching1User];

create user [MediaDataMatching1User] for login [MediaDataMatching1User] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataMatching1User''

if exists (select \* from sys.database\_principals where name = N''MediaDataMatching2User'')

drop user [MediaDataMatching2User];

create user [MediaDataMatching2User] for login [MediaDataMatching2User] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataMatching2User''

if exists (select \* from sys.database\_principals where name = N''MediaDataMatching3User'')

drop user [MediaDataMatching3User];

create user [MediaDataMatching3User] for login [MediaDataMatching3User] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataMatching3User''

if exists (select \* from sys.database\_principals where name = N''MediaDataMatching4User'')

drop user [MediaDataMatching4User];

create user [MediaDataMatching4User] for login [MediaDataMatching4User] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataMatching4User''

if exists (select \* from sys.database\_principals where name = N''MediaDataMatching5User'')

drop user [MediaDataMatching5User];

create user [MediaDataMatching5User] for login [MediaDataMatching5User] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataMatching5User''

if exists (select \* from sys.database\_principals where name = N''MediaDataMatching6User'')

drop user [MediaDataMatching6User];

create user [MediaDataMatching6User] for login [MediaDataMatching6User] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataMatching6User''

if exists (select \* from sys.database\_principals where name = N''MediaDataMatching7User'')

drop user [MediaDataMatching7User];

create user [MediaDataMatching7User] for login [MediaDataMatching7User] with default\_schema = [Media];

exec sp\_addrolemember N''MediaDataBasicAccess'', N''MediaDataMatching7User''

if exists (select \* from sys.database\_principals where name = N''MediaDataMatching1testUser'')

drop user [MediaDataMatching1testUser];

create user [MediaDataMatching1testUser] for login [MediaDataMatching1testUser] with default\_schema = [Media];

exec sp\_addrolemember @rolename = N''MediaDataBasicAccess'', @membername = N''MediaDataMatching1testUser''

if exists (select \* from sys.database\_principals where name = N''MediaDataMatching1rcUser'')

drop user [MediaDataMatching1rcUser];

create user [MediaDataMatching1rcUser] for login [MediaDataMatching1rcUser] with default\_schema = [Media];

exec sp\_addrolemember @rolename = N''MediaDataBasicAccess'', @membername = N''MediaDataMatching1rcUser''

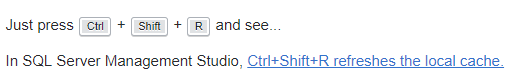
') AT [STOUPA\SQL2016]

### Invalid column name hlaska

Pridal jsem do tabulky novy sloupec a management studio ho podtrhlo cervene a na mouse over zobrazilo tool tip, ze je invalid column name. Sloupec je pritom funkcni a dostupny v select query

Reseni:

Resetnout cache management studiu.



### If column exists - vytvoreni sloupce pokud neexistuje



### Toto cislo verze jiz bylo nasazeno

Problem change scriptu. Pokud jsem vytvoril script , pustil a pak ho upravil, uz ho znovu nepustim .

Reseni:

SELECT top 1 \* from [dbo].[DbSchemaVersion] where [DbSchemaVersion] = 161

DELETE FROM [dbo].[DbSchemaVersion] where [DbSchemaVersion] = 161

1. Sloupec GendeId je nepovinny takze muzeme insertovat tak, ze ho vynechame. Normalne by se do radku v tomto sloupci pridalo NULL. Toto omezeni zaridi, ze když nezadame hodnotu vlozi se defaultni hodnota (3). Pokud je explicitne při insertu vlozeno do sloupce NULL nevlozi se defaultni hodnota ale NULL. [↑](#endnote-ref-1)