**SQL TRAINING**

### Indexy clustered, nonclustered

Indexy

<https://www.youtube.com/watch?v=NGslt99VOCw>

**Clustered and nonClustered index**

**Clustered index**

Urcuje, jak se data v tabulce radi.

Pokud je na tabulce Primary Key, je automaticky vytvoreny clustered index.

Na tabulce muze byt jen jeden clustered index.

Mejme tabulku s primarnim klicem. Kdyz do ni vlozim nekolik radku, s ruznym poradim Id, clustered index nad Id se postara o to, ze kdyz dam Select \* from tbl, dostanu tyto zaznamy serazene podle Id.

Clustered index muze byt na tbl jen jeden, muze byt ale definovany na vice sloupcich. V takovem pripade, mam data v tabulce serazene podle prvniho sloupce, a v pripade shodnych hodnot v tomto sloupci, podle druheho sloupce.

**NonClusteredIndex**

Na tabulce jich muze byt vic a taky muze byt jeden index definovany nad vice sloupcy.

NonClusteredIndex je jako obsah knihy, ve ktere mam cislo kapitoly ve ktere je hledany obsah.

NonClusteredIndex vytvori dalsi tabulku, ve ktere je serazeny indexovany sloupec (sloupce) a k tomu je prirazena adresa radku v puvodni tabulce, kde se tento radek nachazi.

**Vytvoreni indexu:**

create nonclustered index nonLastName on dbo.Customer(LastName)

Vytvori index s nazvem nonLastName, nad sloupcem LastName v tabulce Customer.

### Exekucni plany, indexy, optimalizace, ready

Exekucni plany:

<https://d.docs.live.net/b22fb0fb09218bf0/Nielsen%20%20prace/Moje%20poznamky%20Nielsen/SQL%20Execution%20plans.docx>

### AdoNet pripojeni k databazi psano s Mitrozem ve vesmiru

(OneDrive\ Nielsen Prace\ AdoNet Konec) Mame tridu Entity napr Galaxie, ktera ma property podle database. Dale mame GalaxieDataContract ktery krome propert podle entity ma navic napr metodu Create, ktera vrati novy dataContract.

Pro pristup do database mame napr pro entitu Galaxie tridu GalaxyDao : EntityDaoBase. Pokud chceme neco z databaze dostat, musime metode LoadEntity predat instanci readeru a query string. Kdyz chceme jen neco provest na db, ale nic nechceme vratit, staci query a parametry.

namespace WpfUniverse.Entities

{

public abstract class EntityDaoBase<TEntity>

{

protected EntityDaoBase(string connectionString)

{

ConnectionString = connectionString; //do protected promenne priradime connectionString

}

protected string ConnectionString { get; }

/// vraci list of entity, jmenuje se LoadEntity, jako parametr si bere "nejakou entitu" a jako parametr dostane dotaz a SqlDataReader ,ktery uz ma nactenou entitu i s vlastnostmi

/// parametry jsou null tj nepovinne

/// <typeparam name="TEntity">jakakoli entita napr galaxie nebo planeta</typeparam>

/// <param name="query">je sql dotaz ve kterem urcujeme jake vlastnosti budeme z databaze potrebovat je definovany ve zdedenych tridach</param>

/// <param name="factoryMethod">metoda definovana ve zdedenych tridach vytvori objekt podle tridy ve ktere je volane a priradi mu vlastnosti vybrane v query</param>

/// <returns>vraci seznam entit podle toho ve ktere zdedene tride je metoda volana</returns>

protected List<TEntity> LoadEntity(string query, Func<SqlDataReader, TEntity> factoryMethod, SqlParameter[] parameters = null)

{

List<TEntity> result = new List<TEntity>();

using (SqlConnection conn = new SqlConnection(ConnectionString))

{

SqlTransaction transaction = null;

try

{

conn.Open();

transaction = conn.BeginTransaction();

using (SqlCommand command = new SqlCommand(query, conn, transaction))

{

if (parameters != null && parameters.Length > 0)

{

command.Parameters.AddRange(parameters);

}

using (SqlDataReader reader = **command.ExecuteReader())**

{

while (reader.Read())

{

var vlp = factoryMethod.Invoke(reader);

result.Add(vlp);

}

}

}

transaction.Commit(); // Neco jako Submit

}

catch (Exception ex)

{

transaction?.Rollback(); // Vrat se zpatky a neprovadej zadne zmeny

Console.WriteLine(ex);

throw;

}

finally

{

conn.Close();

}

}

return result;

}

protected int ExcuteInsert(string query, SqlParameter[] parameters = null)

{

using (SqlConnection conn = new SqlConnection(ConnectionString))

{

SqlTransaction transaction = null;

try

{

conn.Open();

transaction = conn.BeginTransaction();

int id;

using (SqlCommand command = new SqlCommand(query, conn, transaction))

{

if (parameters != null && parameters.Length > 0)

{

command.Parameters.AddRange(parameters);

}

id = Convert.ToInt32(**command.ExecuteScalar());**

}

transaction.Commit();

return id;

}

catch (Exception ex)

{

transaction?.Rollback();

Console.WriteLine(ex);

throw;

}

finally

{

conn.Close();

}

}

}

protected void ExcuteUpdate(string query, SqlParameter[] parameters = null)

{

using (SqlConnection conn = new SqlConnection(ConnectionString))

{

SqlTransaction transaction = null;

try

{

conn.Open();

transaction = conn.BeginTransaction();

using (SqlCommand command = new SqlCommand(query, conn, transaction))

{

if (parameters != null && parameters.Length > 0)

{

command.Parameters.AddRange(parameters);

}

command.ExecuteNonQuery();

}

transaction.Commit();

}

catch (Exception ex)

{

transaction?.Rollback();

Console.WriteLine(ex);

throw;

}

finally

{

conn.Close();

}

}

}

}

}

### Command.ExecuteReader()

Potrebujeme instance readeru, ktery cte data z database

using (SqlDataReader reader = command.ExecuteReader())

{

while (reader.Read())

{

var vlp = factoryMethod.Invoke(reader);

result.Add(vlp);

}

}

Sends the SqlCommand.CommandText to the SqlCommand.CommandConnection and builds the SqlDataReader

### command.ExecuteScalar()

Vrati data z databaze (Select)

id = Convert.ToInt32(command.ExecuteScalar());

vraci prvni sloupec prvního radku result setu, který vrati query. Ostatní radky a sloupce jsou ignorovane.

### command.ExecuteNonQuery();

Provede cinnost na serveru a nic nevraci (Update)

Executes the T-Sql statement against the connection and returns the number of rows affected

### 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 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****

|  |  |
| --- | --- |
| **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 |
|  |  |



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

### Update

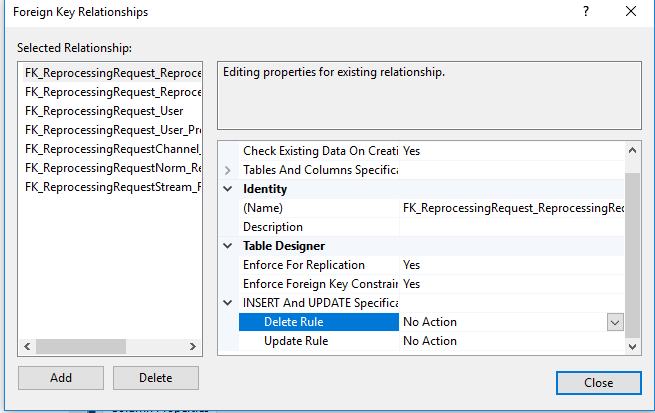
UPDATE table\_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;

### [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)

### ****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

### 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)

### 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.

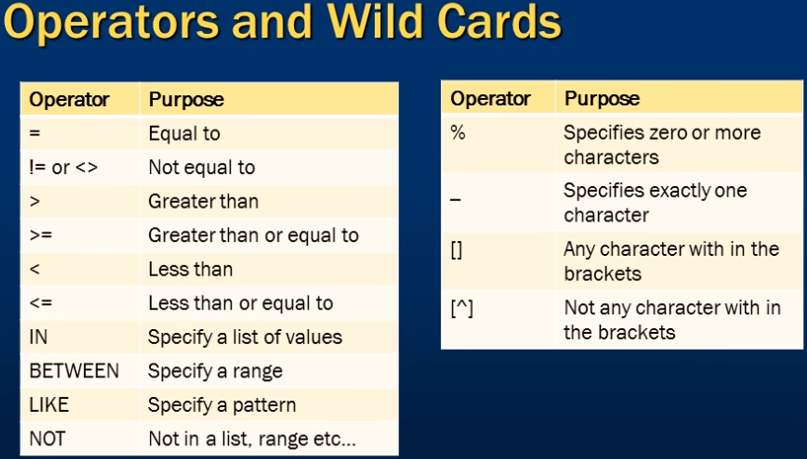
SELECT DISTINCT Name, City FROM Person

**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

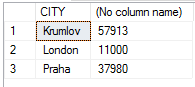
SELECT MAX(SALARY) FROM PERSON

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

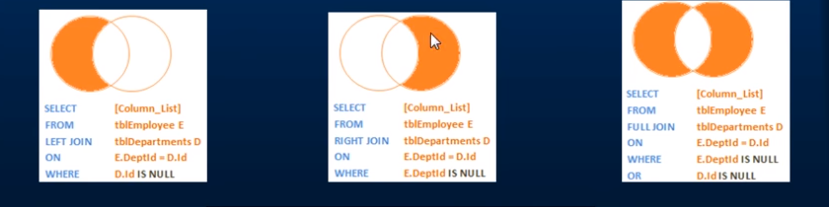
<https://www.youtube.com/watch?v=GKGtOABAO9s>

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:**

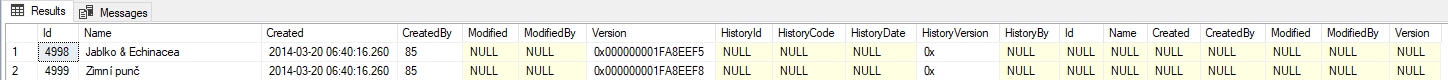
select \* from media.ProductDetail pro

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

****

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

<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

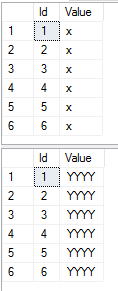
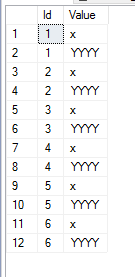
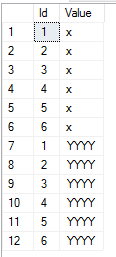
<https://www.youtube.com/watch?v=9w5uRCFOiTo>

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

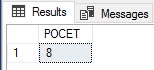
SELECT @TotalCount = COUNT(Id) from dbo.Person

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

### 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>

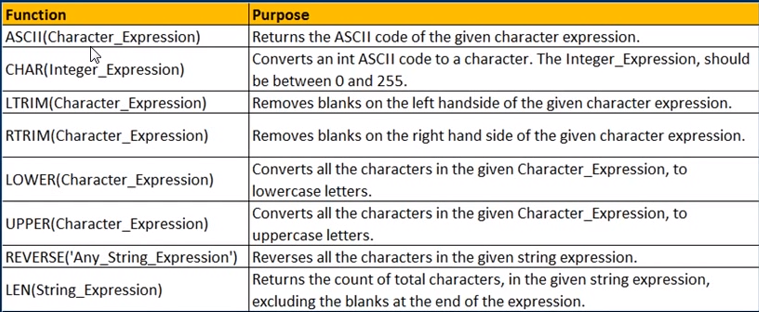
Nazev databaze -> Programmability -> Functions -> System Functions -> String Functions

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

(

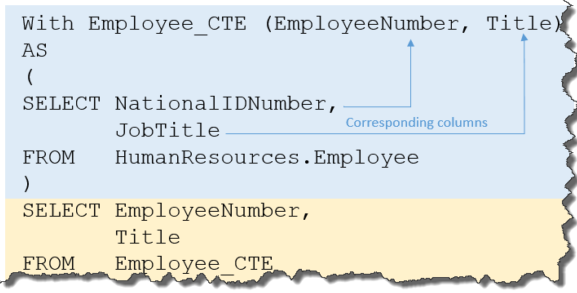
Select \*, ROW\_NUMBER () Over(Partition by Id order by Id) as RowNumber

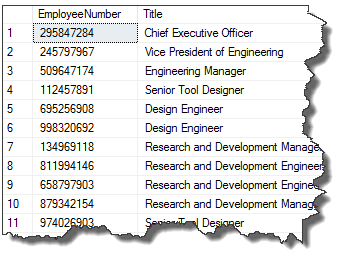
from ##Pricing\_TVMessageIds

)

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

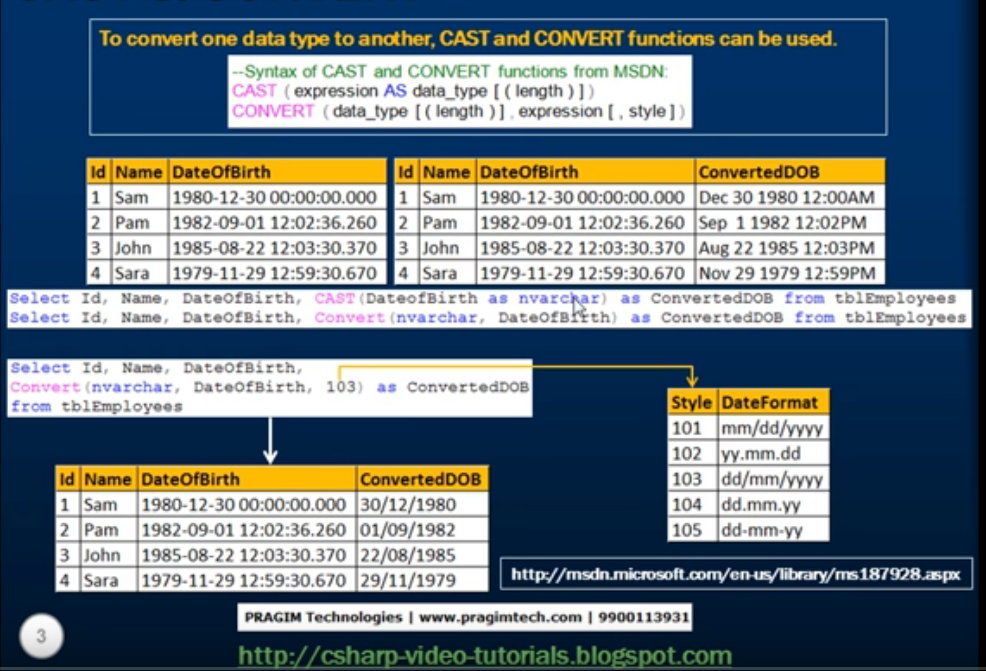
)

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 :**

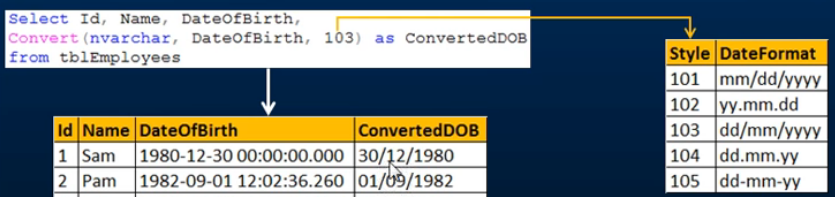








**Convert (style) k dispozici jsou preddefinovane styly:**

****

### 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'),

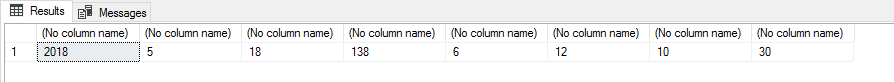
DATEPART(DAYOFYEAR, '2018-5-18 12:10:30.123'),

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

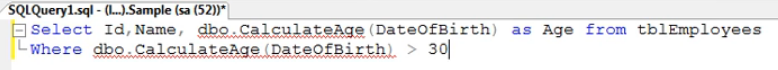
return @Age

end

**Pouziti:**

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

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

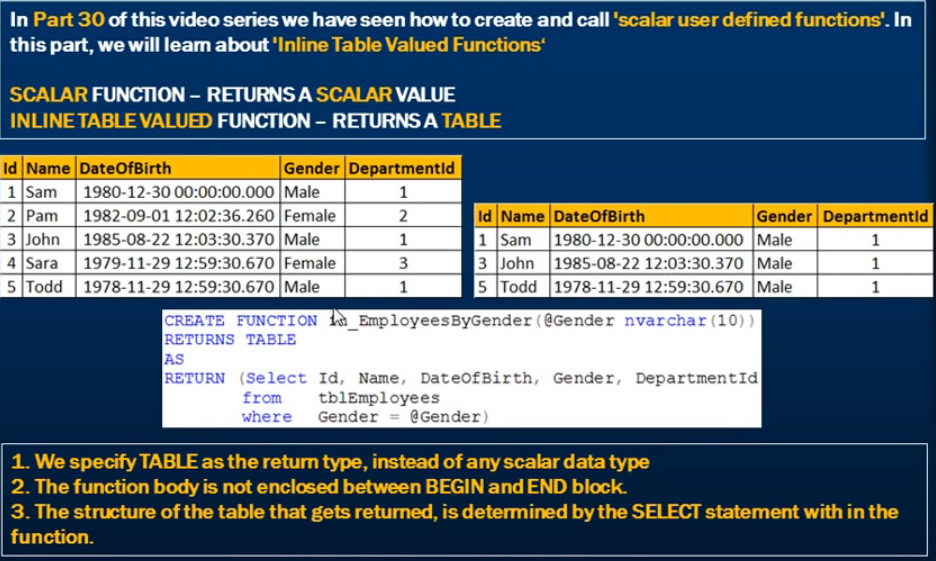


To ze je to podtrzene nijak nevadi, jen se musi zadat nazev databaze a nazev funkce. Muze tam byt mezi dbo.

### 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()

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 --

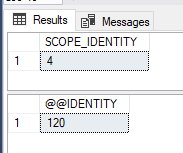
INSERT A VALUES ('Rosalie');

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

GO

SELECT @@IDENTITY AS [@@IDENTITY];

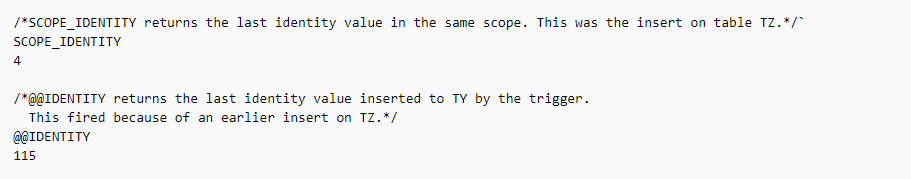
GO



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>



### 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)

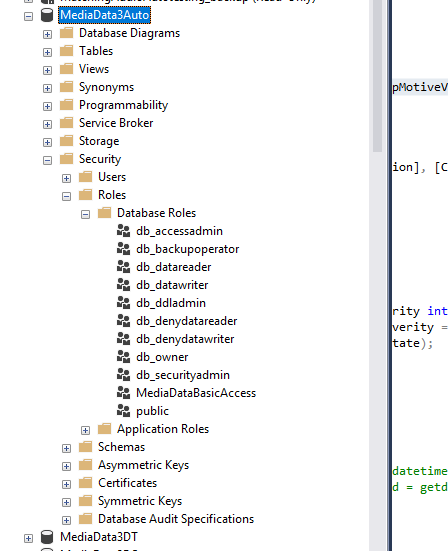
**BLToolkit.Data.DataException:** 'The SELECT permission was denied on the object 'TvImportItem', database 'MediaData3Auto', schema 'Import'.

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

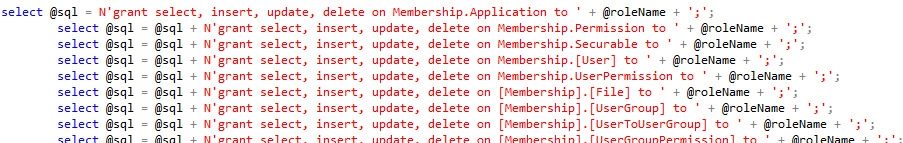
grant jde přidat i pro konrétní uživatele, ale takhle to mají rovnou všichni

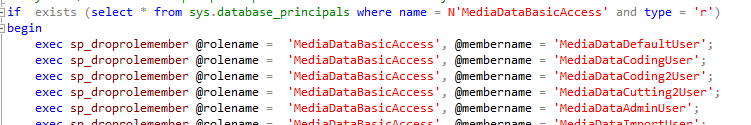
[‎6/‎28/‎2018 4:43 PM] Peter Hlavenka:

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:**

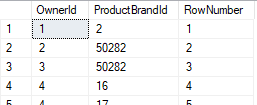
Pokud v over casti necham jen OrderBy, dostanu ve sloupci RowNumber jen serazene radky od 1.

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.

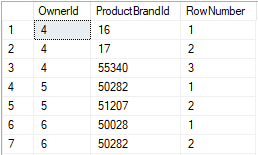
SELECT TOP(100)

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)

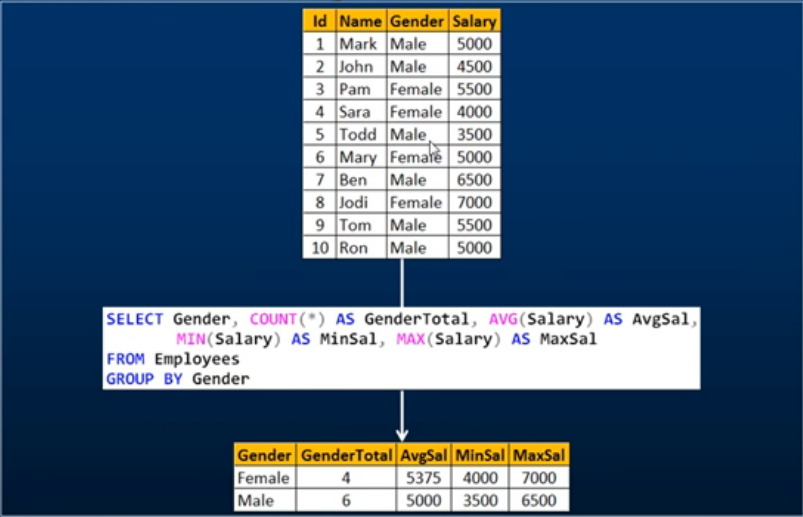
### Over clause

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

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

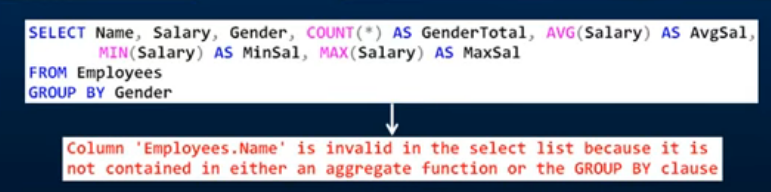
Tohle resi problem GroupBy, kde muzu dostat pro nejakou kategorii pocet radku, ktere splnuji group klauzuli

Priklad : Pomoci group by dostaneme prumernou mzdu pro kazde pohlavi

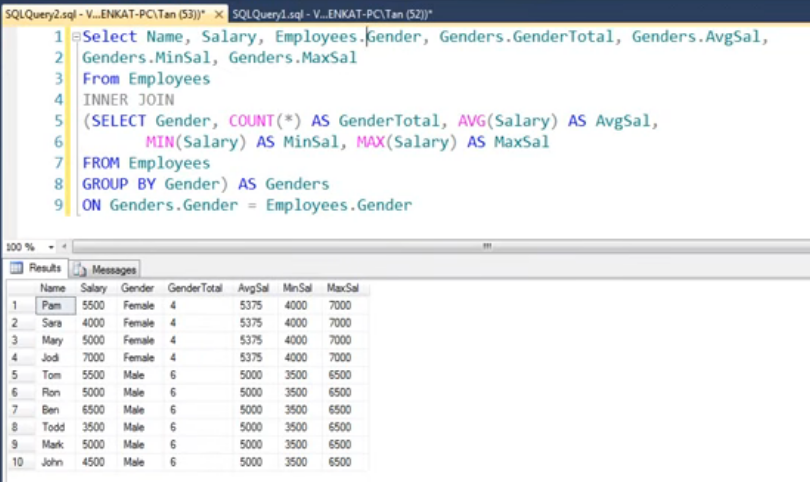


Co kdybych chtel tyto pohlavi vypsat radek po radku a v kazdem bych chtel videt prumernou mzdu pro dane pohlavi?

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

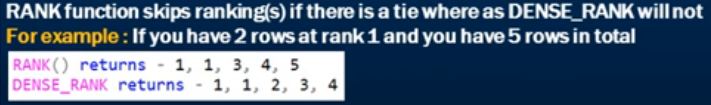
|  |  |
| --- | --- |
| *podstatné jméno* | pořadí, hodnost, postavení, řada, stupeň, řád, linie |

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.

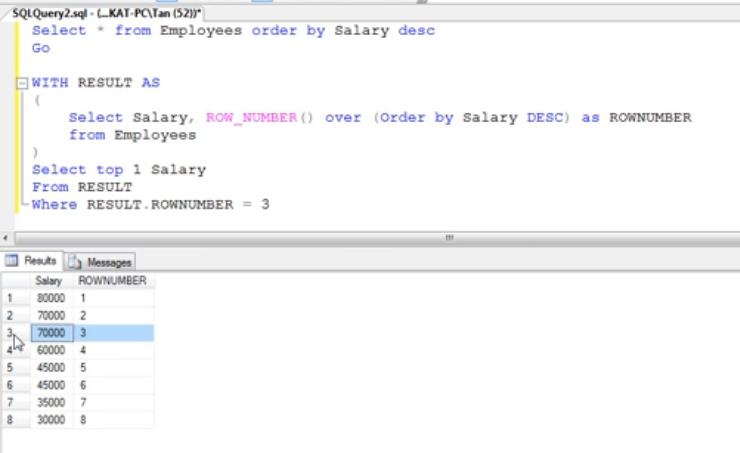
**Pouziti:**

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

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

### Ziskani IP Adresy serveru pomoci query

SELECT dec.local\_net\_address

FROM sys.dm\_exec\_connections AS dec

WHERE dec.session\_id = @@SPID;

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)