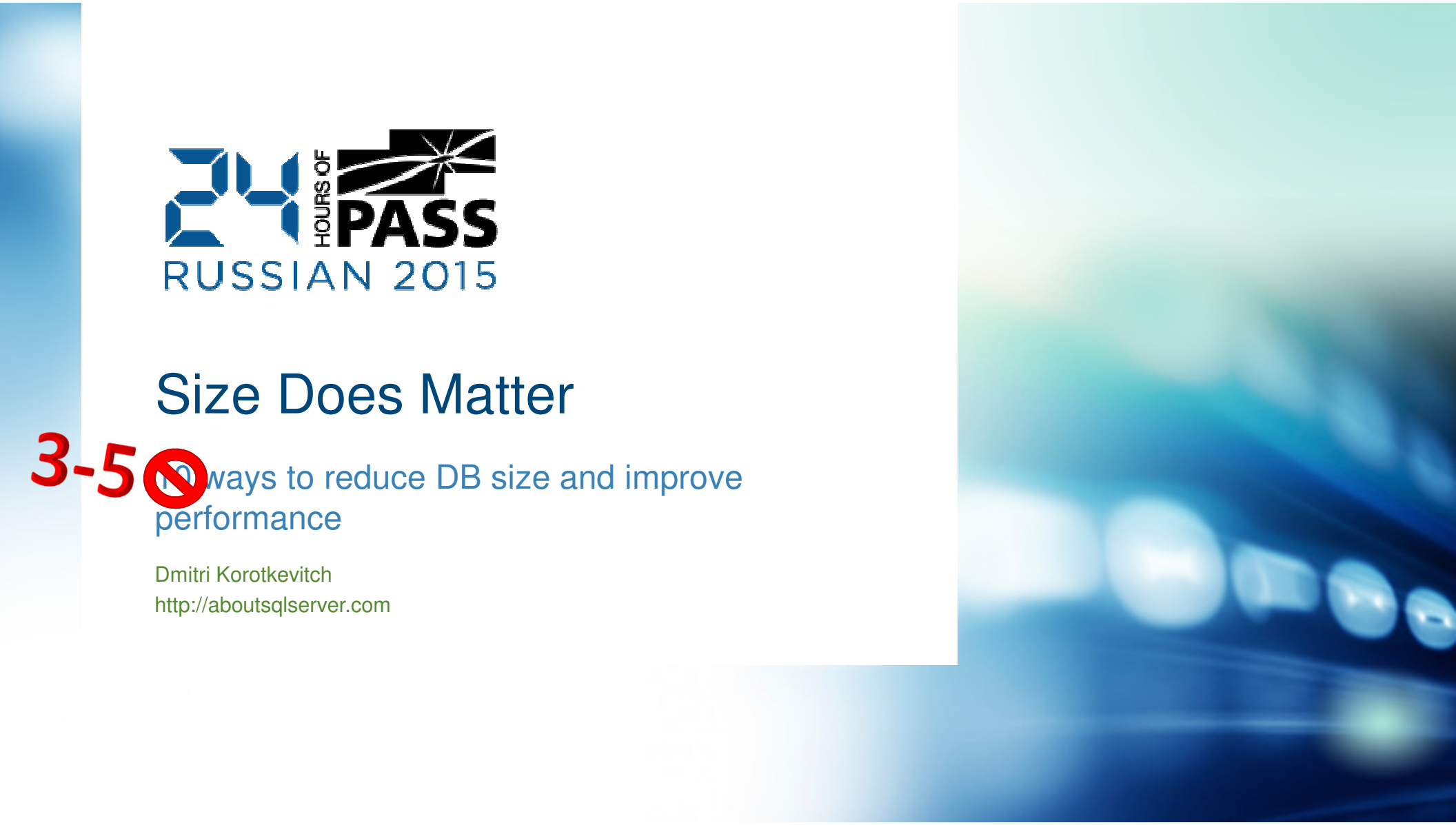



# Size Does Matter

3-5

## 10 ways to reduce DB size and improve performance

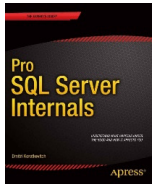
Dmitri Korotkevitch
http://aboutsqlserver.com

# About me

20+ years of experience in IT
15+ years of experience working with SQL Server


Microsoft SQL Server MVP
Microsoft Certified Master (SQL Server 2008)
Author of “Pro SQL Server Internals”


Blog: http://aboutsqlserver.com
Email: dk@aboutsqlserver.com



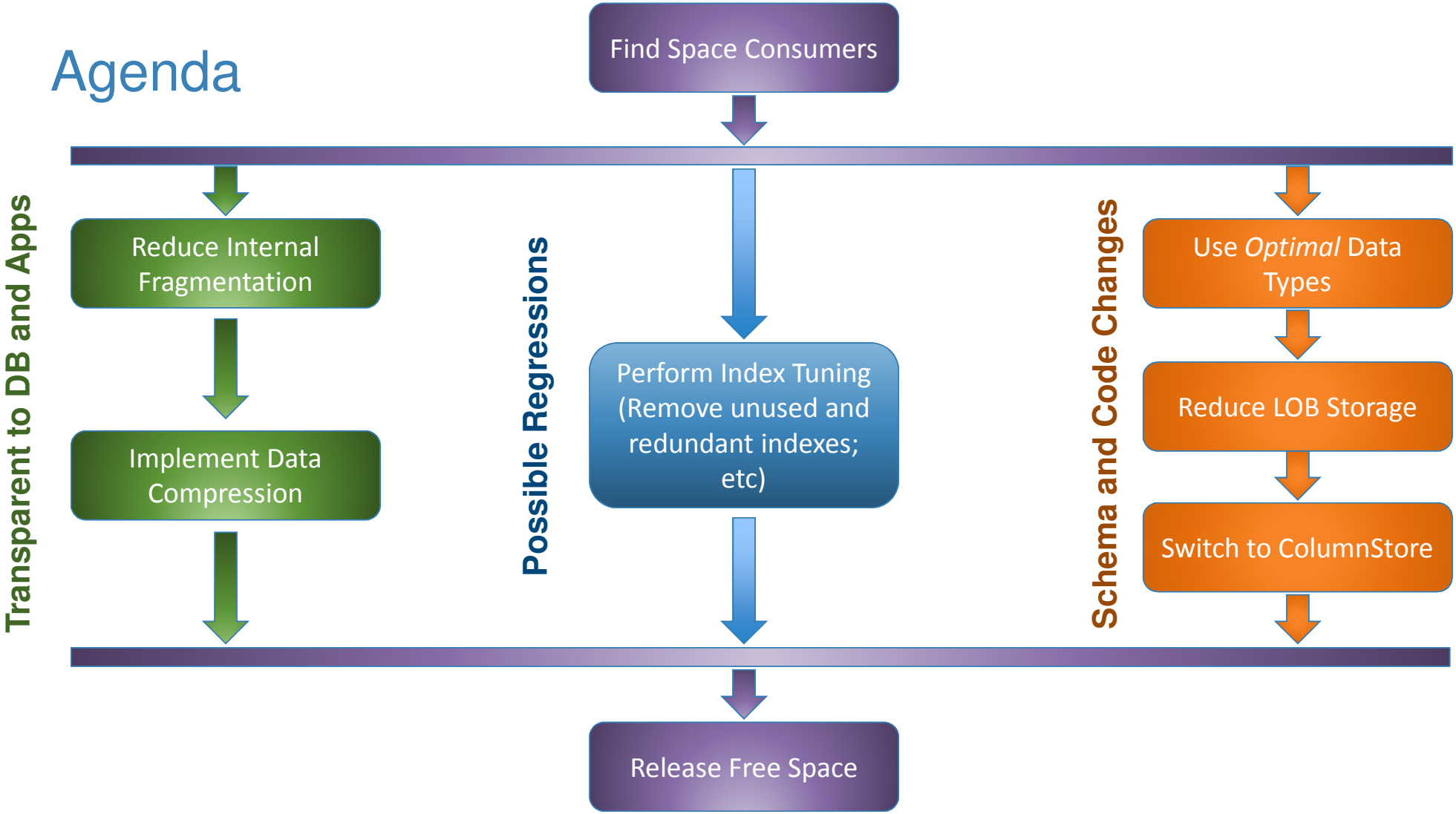
# Why bother reducing DB Size?

- Improves Performance
- Simplifies SLA, RTO, RPO Compliance
- Makes Administration and Maintenance Easier
- Reduces Hardware Cost

Picture from: https://openclipart.org/detail/172012/award-by-gnokii-172012


# Demo:
# DBCC SMART_SHRINK()

# Agenda


Find Space Consumers
Transparent to DB and Apps
Reduce Internal Fragmentation
Implement Data Compression
Possible Regressions
Perform Index Tuning (Remove unused and redundant indexes; etc)
Schema and Code Changes
Use *Optimal* Data Types
Reduce LOB Storage
Switch to ColumnStore
Release Free Space

# Best Practices

## Auto Shrink is OFF

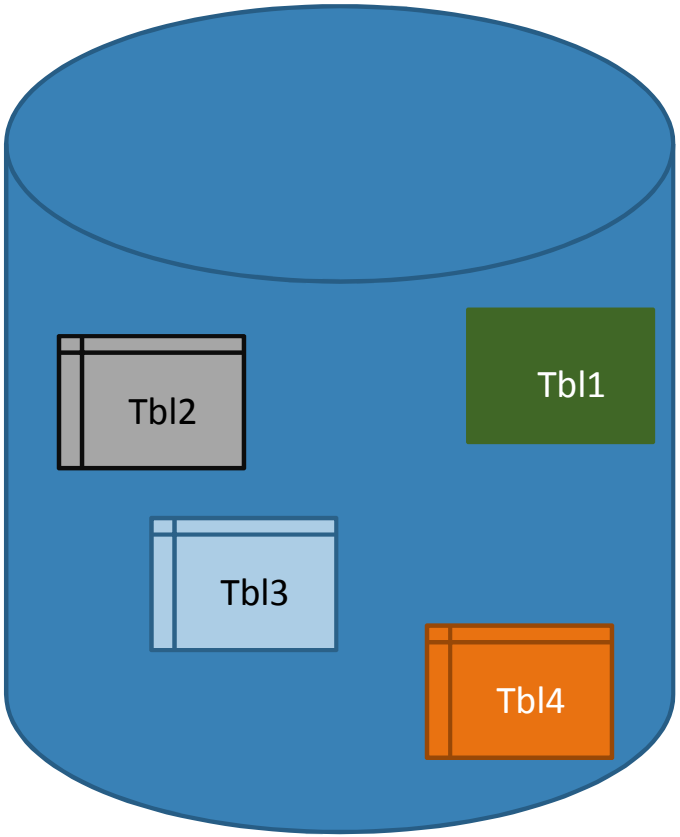
- Greatly increases index fragmentation
- Contributes to I/O, CPU load and T-Log overhead
- Absolutely useless

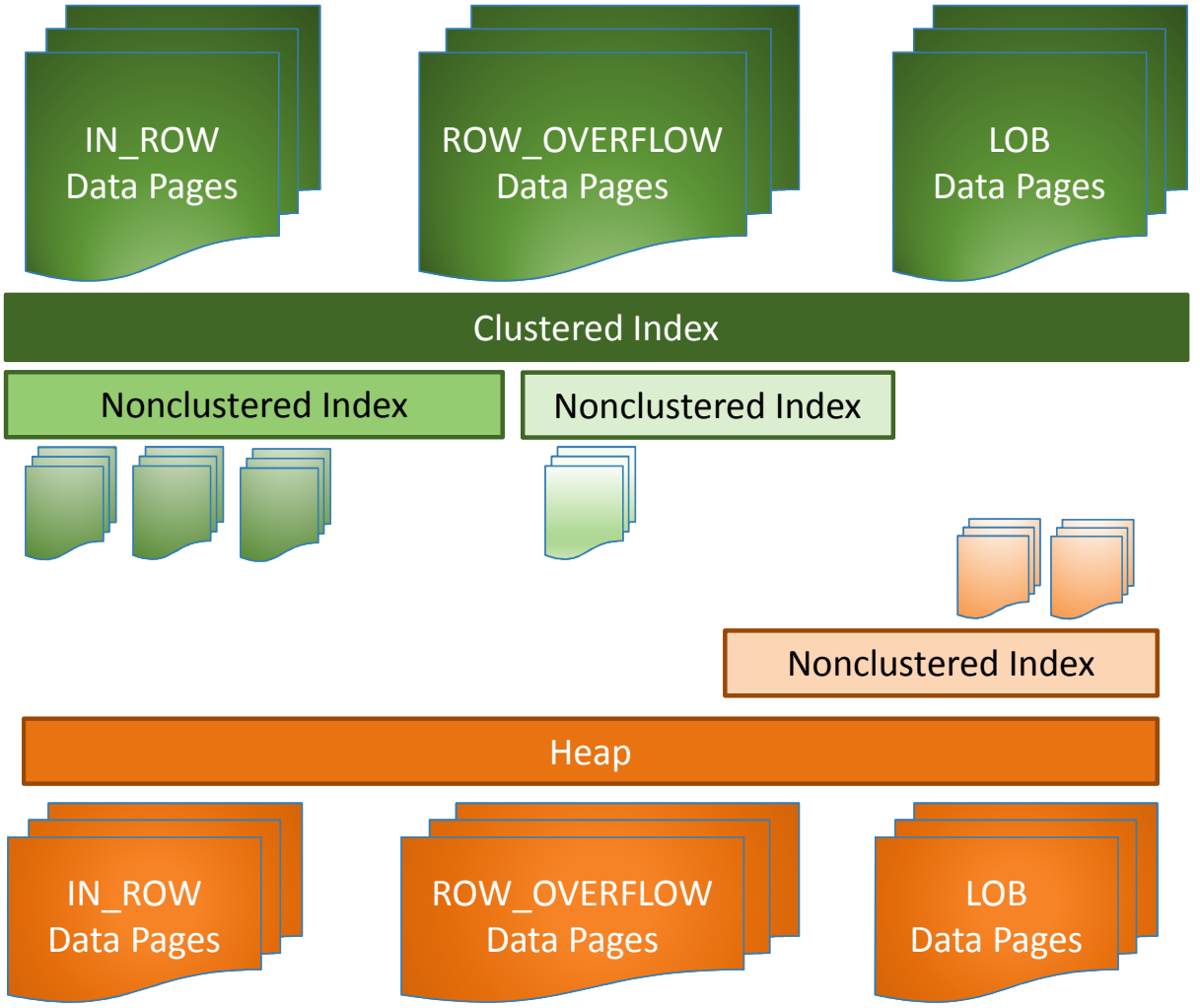
## Instant File Initialization is ON

- Prevents zeroing-out of the newly allocated space in the data files
- Speed-up data file (auto)-growth and database restore
- Does not affect T-Log growth (it is always zeroing out)


# Demo: Instant File Initialization

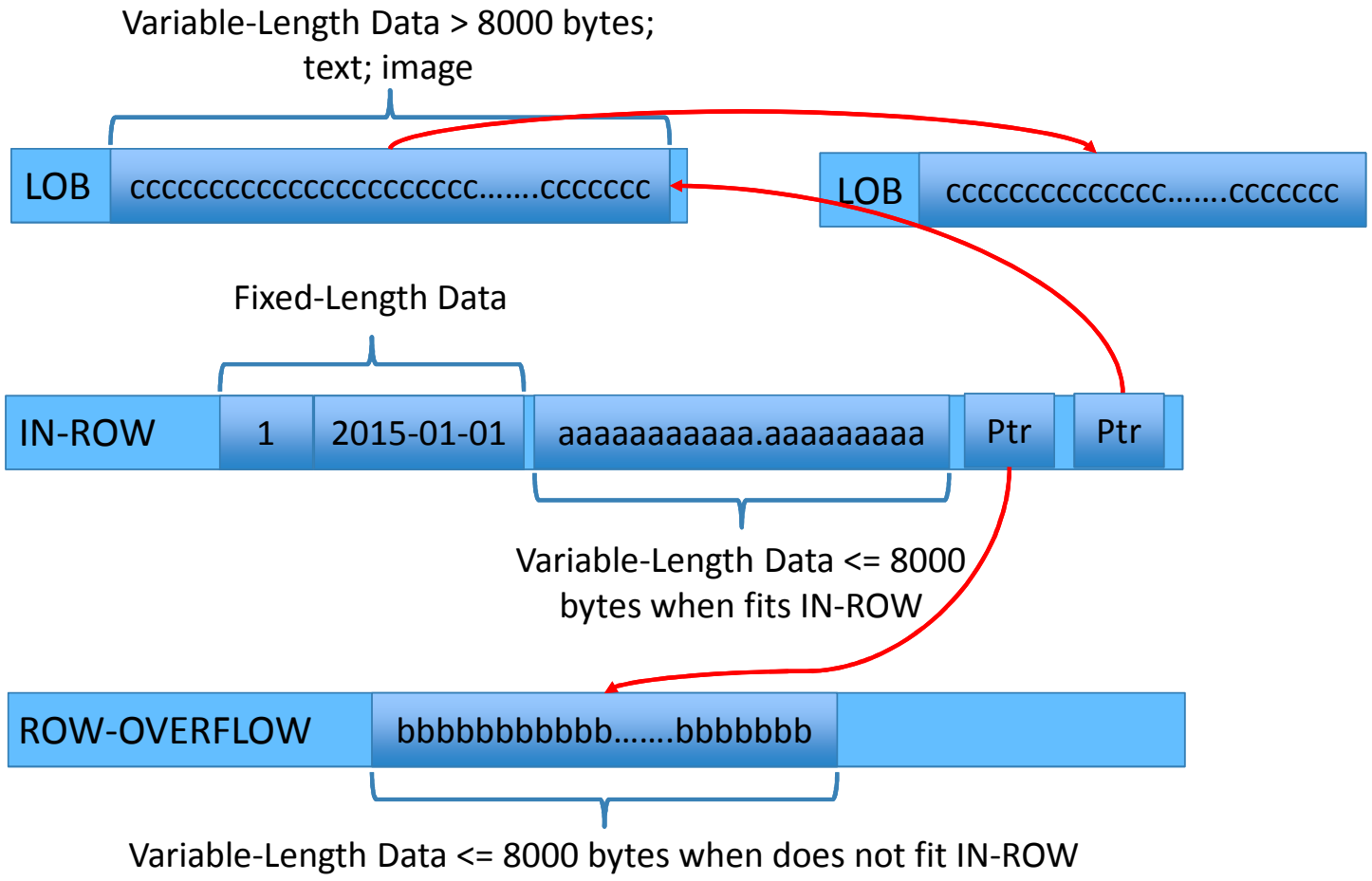
# Data Storage 101



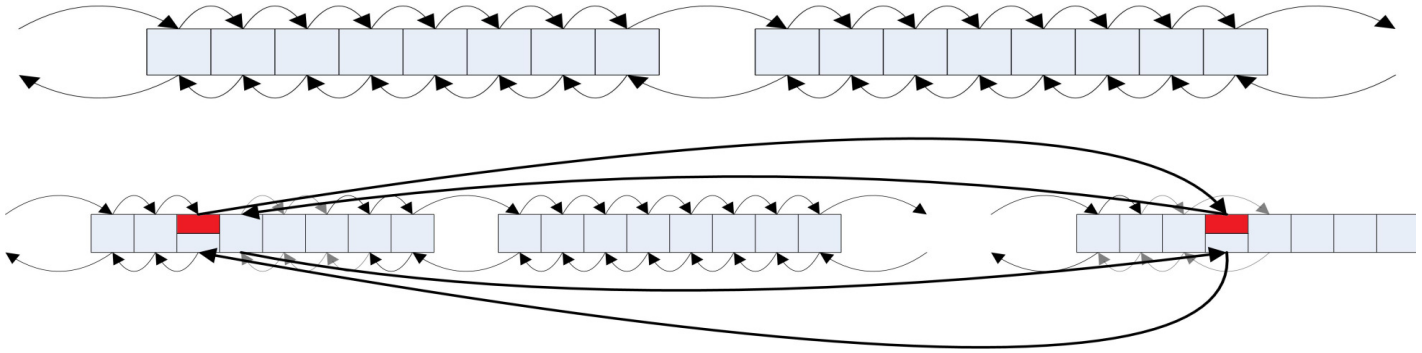
```
create table T
(
   ID int,
   DateCol datetime,
   VarCol1 varchar(5000),
   VarCol2 varchar(5000),
   VarCol3 varchar(max)
);

insert into T values
(
   1
   ,'2015-01-01'
   ,REPLICATE('a',5000)
   ,REPLICATE('b',5000)
   ,REPLICATE('c',32000)
);
```

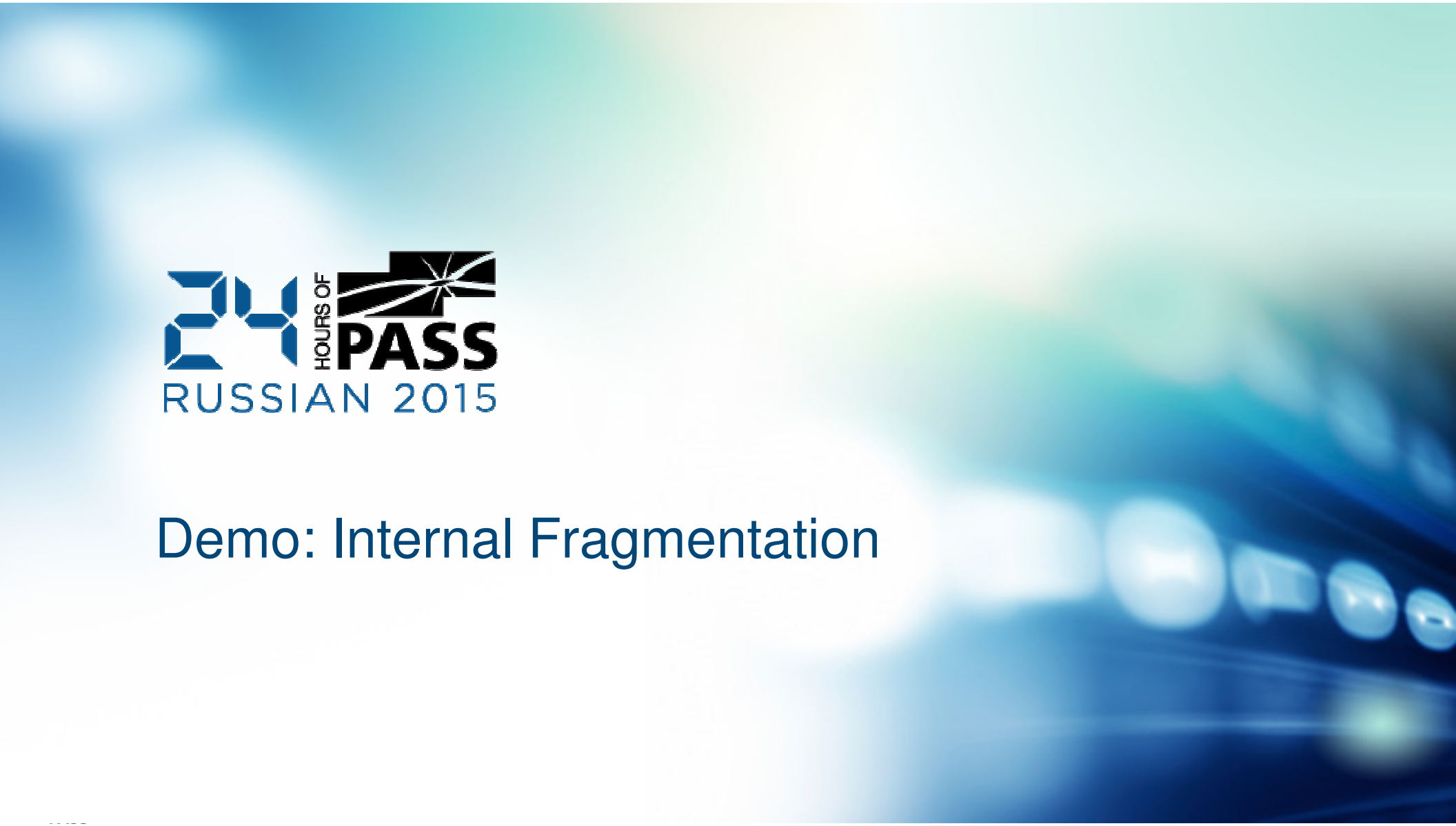
# Data Storage 101




# Demo: Find Space Consumers

# Internal Fragmentation






# Demo: Internal Fragmentation

# Data Compression (Enterprise Edition)

## Uncompressed Row

- Fixed-Length Columns: storage space is based on the data type regardless of the value
- Variable-Length Columns: storage space is based on the data size + 2 bytes

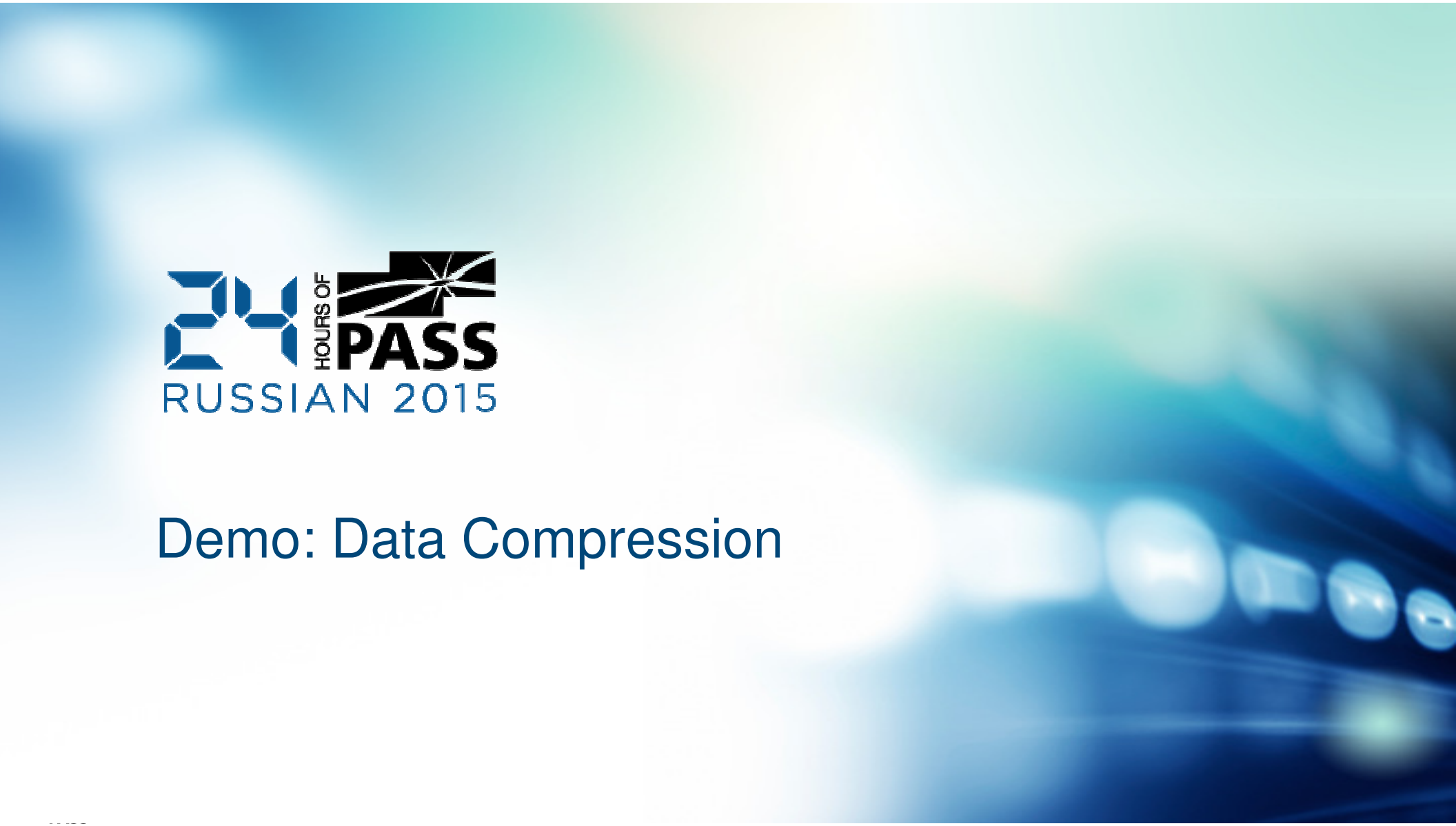
## ROW Compression

- Reduces fixed-length columns storage overhead

## PAGE Compression = ROW Compression + Prefix + Dictionary Compression

- Works on the data page scope

Both, PAGE and ROW compression works only with IN_ROW data

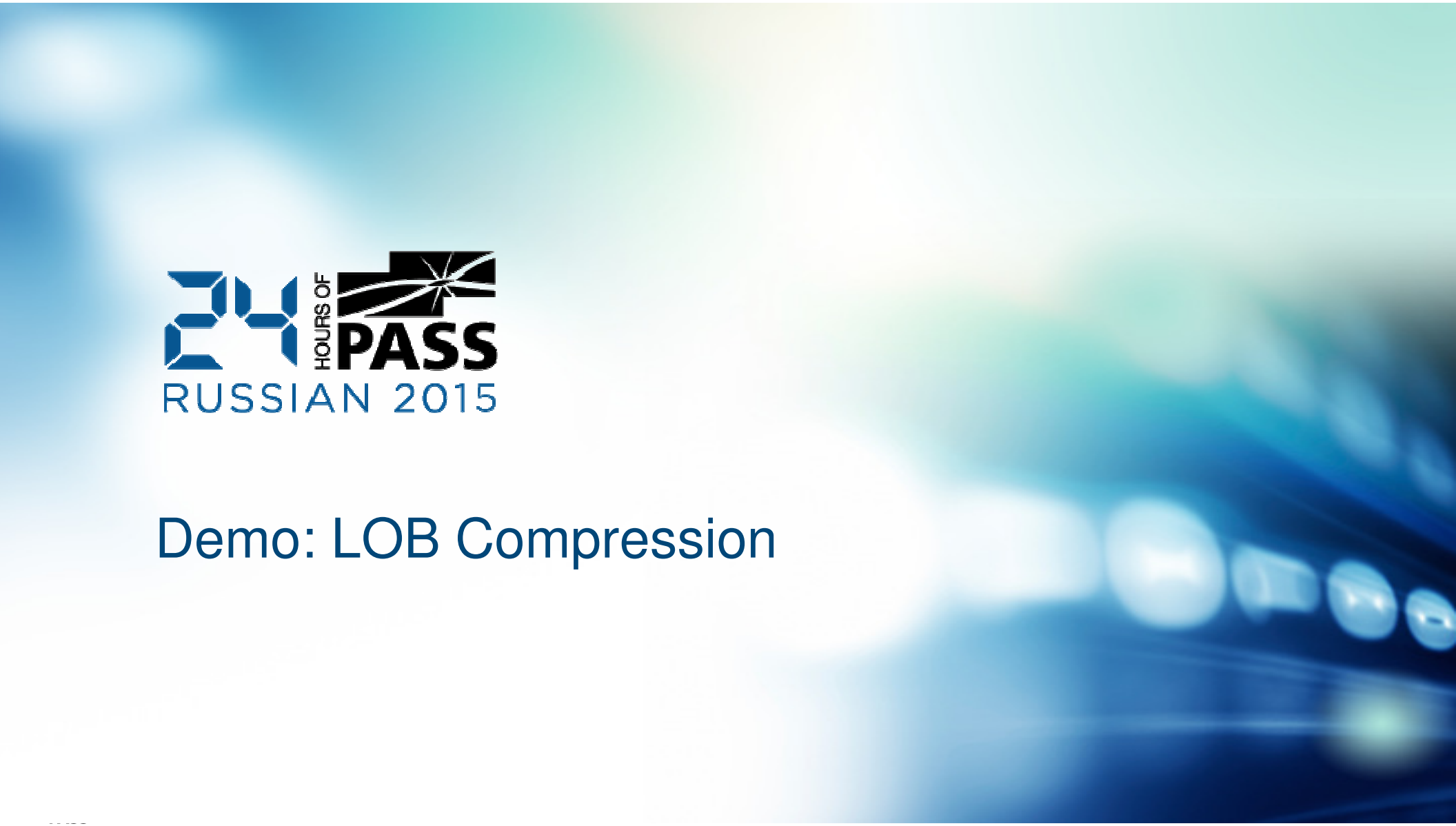



# Demo: Data Compression

# LOB Compression

```
[Microsoft.SqlServer.Server.SqlFunction
    (IsDeterministic = true, IsPrecise = true,
        DataAccess = DataAccessKind.None)]
public static SqlBytes BinaryCompress(SqlBytes input)
{
     if (input.IsNull)
         return SqlBytes.Null;

    using (MemoryStream result = new MemoryStream())
    {
        using (DeflateStream deflateStream = new DeflateStream(result,
            CompressionMode.Compress, true))
        {
            deflateStream.Write(input.Buffer, 0, input.Buffer.Length);
            deflateStream.Flush();
            deflateStream.Close();
        }
        return new SqlBytes(result.ToArray());
    }
}
```

```
[Microsoft.SqlServer.Server.SqlFunction
    (IsDeterministic = true, IsPrecise = true,
        DataAccess = DataAccessKind.None)]
public static SqlBytes BinaryDecompress(SqlBytes input)
{
    if (input.IsNull)
        return SqlBytes.Null;

    int batchSize = 32768;
    byte[] buf = new byte[batchSize];

    using (MemoryStream result = new MemoryStream())
    {
        using (DeflateStream deflateStream =
                new DeflateStream(input.Stream,
                    CompressionMode.Decompress, true))
        {
            int bytesRead;
            while ((bytesRead = deflateStream.Read(buf, 0, batchSize)) > 0)
                result.Write(buf, 0, bytesRead);
        }
        return new SqlBytes(result.ToArray());
    }
}
```




# Demo: LOB Compression

# Use Appropriate Data Types

## Obvious Mistakes:

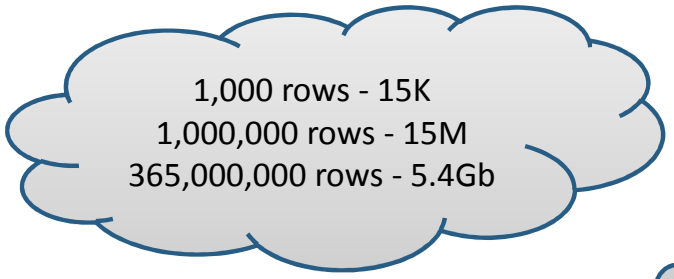
- Boolean -> tinyint, smallint, int
- (n)char(N)

## What kind of precision do you need?

- DateTime -> SmallDateTime or DateTime2
- Float -> Decimal

## Do not use LOB data types unless you need them

# Use Appropriate Data Types




```
create table Positions
(
    ATime datetime not null, -- 8 bytes
    Latitude float not null, -- 8 bytes
    Longitude float not null, -- 8 bytes
    IsValid int not null, -- 4 bytes
    IsAssistanceUsed int not null, -- 4 bytes
    -- total: 32 bytes
)
```

```
create table Positions2
(
    ATime datetime2(0) not null, -- 6 bytes
    Latitude decimal(9,6) not null, -- 5 bytes
    Longitude decimal(9,6) not null, -- 5 bytes
    IsValid bit not null, -- 1 byte
    IsAssistanceUsed bit not null, -- 0 bytes
    -- total: 17 bytes
)
```

Think about future and do not be cheap

Rebuild indexes after table alteration


# Demo: Alteration and Data Row Size

# Index Tuning: Removing Unused Indexes

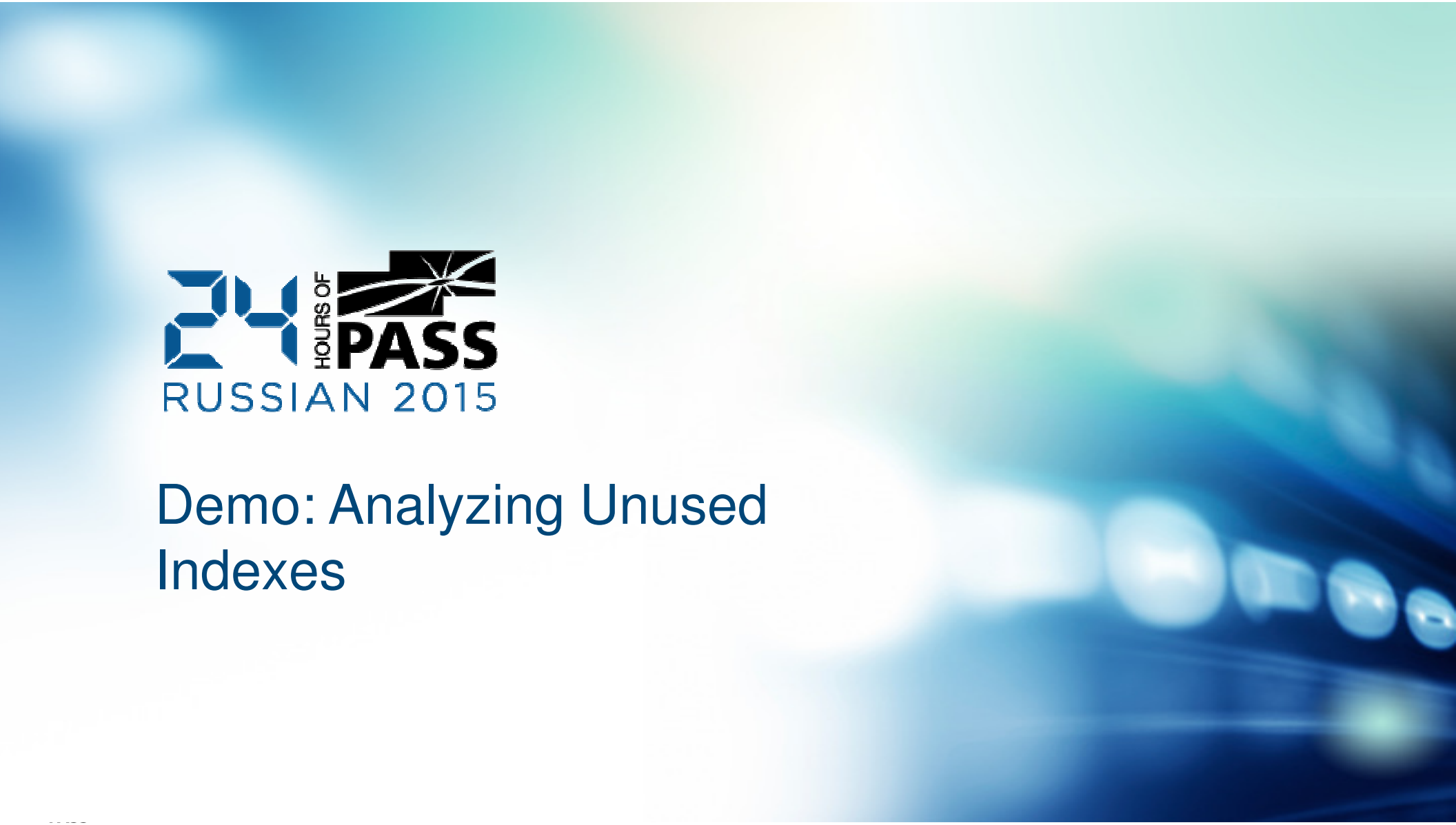
## sys.dm_db_index_usage_stats

- How many times index appear in the execution plan

## sys.dm_db_index_operational_stats

- How many times operation occurs
- Includes I/O, Locking and Latching statistics

Statistics clears on SQL Server restart and index rebuild (SQL Server 2012+)




# Demo: Analyzing Unused Indexes

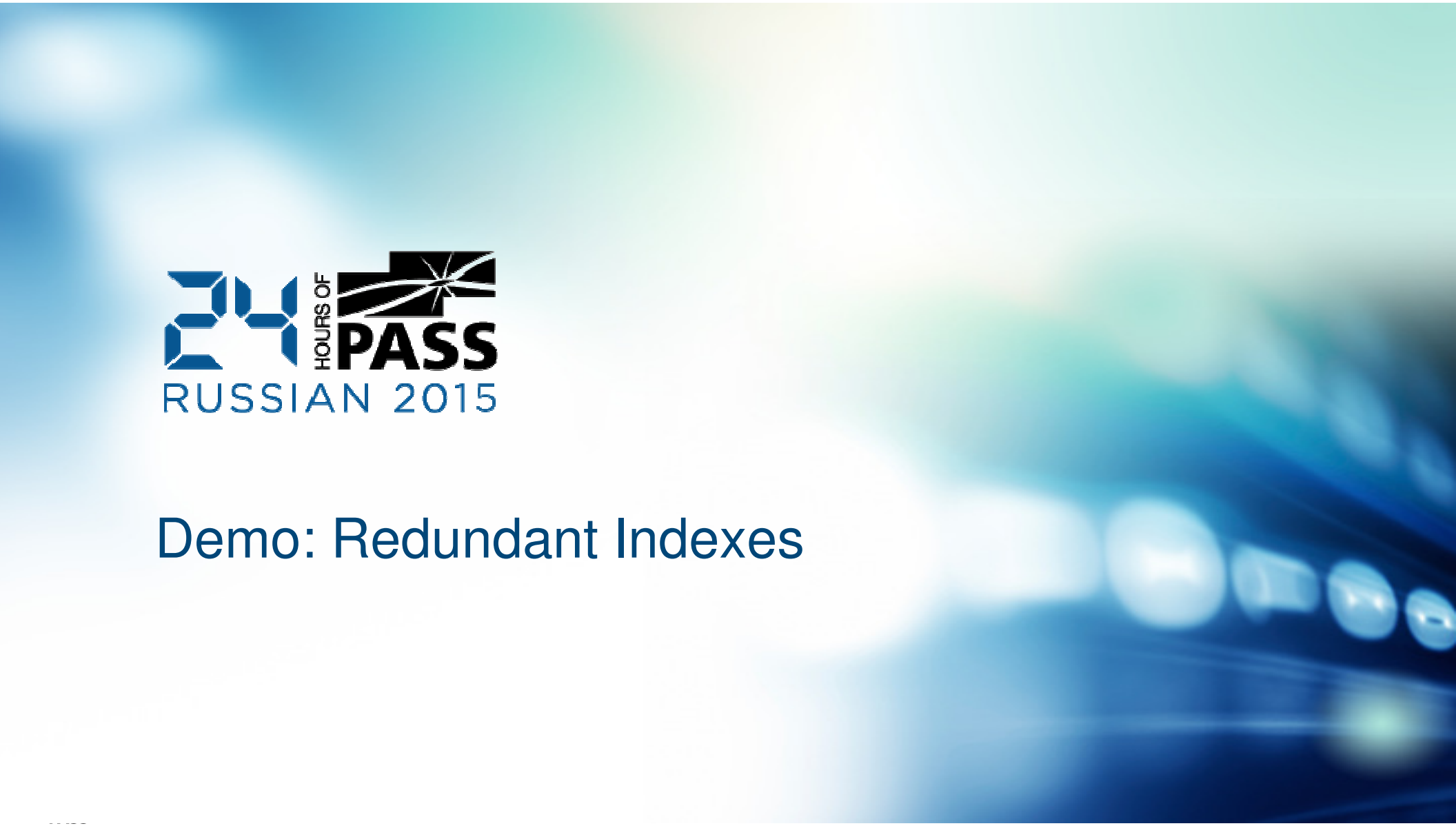
# Index Tuning: Detecting Redundant Indexes

IDX1(A, B) & IDX2(A) -> IDX2 is redundant

IDX3(A) INCLUDE (B) & IDX4(A) INCLUDE (C) -> IDX5(A) INCLUDE(B,C)

IDX6(A,B) & IDX7(A,C) ->

- IDX8(A,B) INCLUDE (C)
- IDX9(A,C) INCLUDE (B)
- Or keep IDX6 and IDX7




# Demo: Redundant Indexes

# Releasing Empty Space

## DBCC SHRINKFILE

- Introduces Fragmentation (consider to REORG indexes afterwards)
- Challenging in case of multiple files in the filegroup

## INSERT INTO NewTable SELECT FROM OldTable

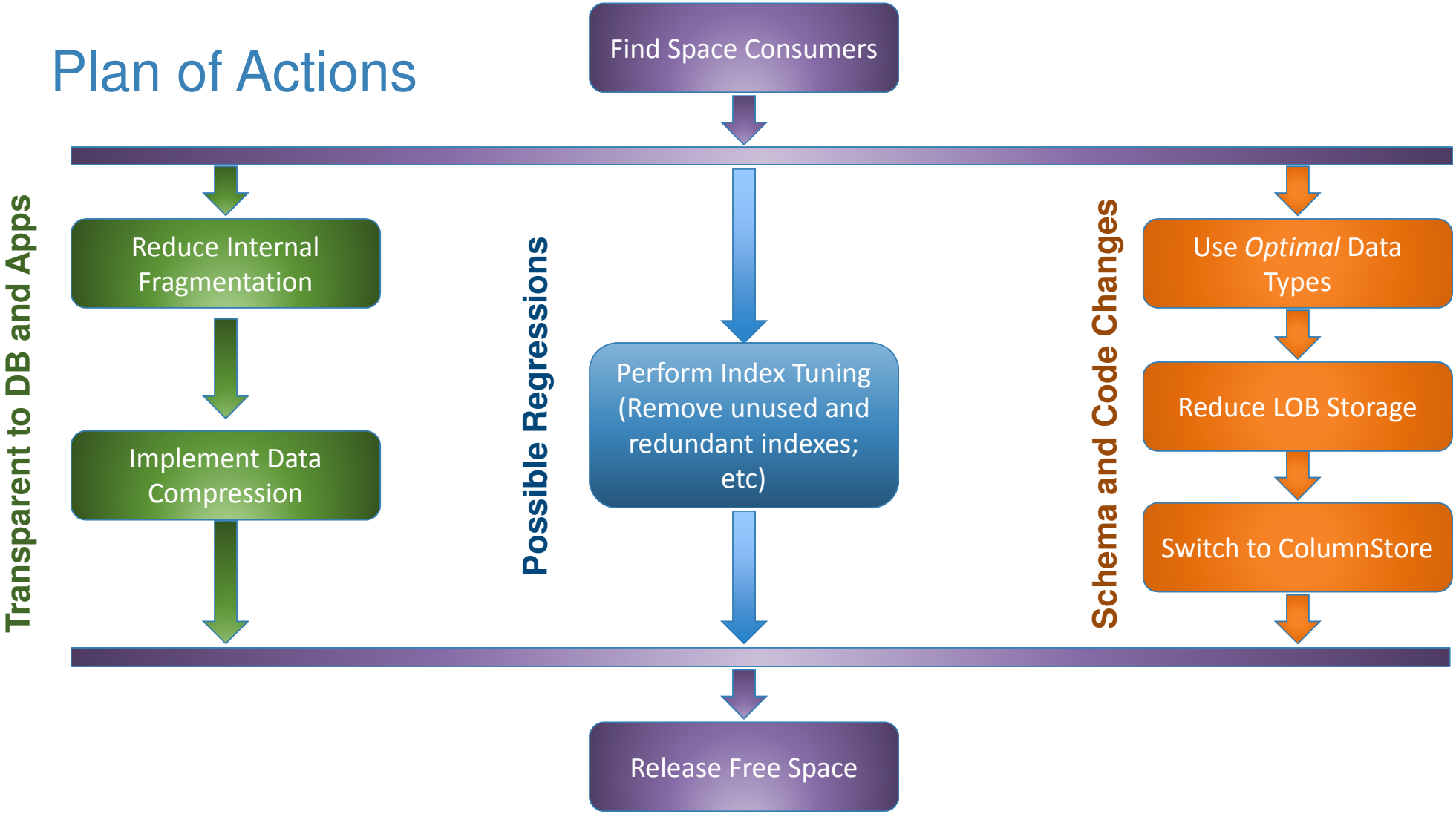
- Offline Operation

## CREATE INDEX WITH (DROP_EXISTING=ON) ON [NewFileGroup]

- Issues with LOB Data


# Demo: Moving Data to New FileGroup

# Plan of Actions



# Q&A

Thank you very much for attending!

Blog: http://aboutsqlserver.com

- Scripts are available for download: http://aboutsqlserver.com/presentations
- Recent post: “Size Does Matter..”: http://goo.gl/J644Zz

E-Mail: dk@aboutsqlserver.com