Modern SQL Server Features That Make Life Better

Thank You!

Thank you, GroupBy Conference!!

About Me

- Entered IT in Feb 2008
- MCTS SQL Server 2008 Implementation and Maintenance in May 2013
- Database Administrator since 2014
- Blog at https://leemarkum.com/
- Speaking: Meetups, New Stars of Data in 2021, online for Pass in 2022, SQL Saturday Columbus in 2023, and Saint Louis DevUp in 2024
- Currently a Senior Database Administrator at Rocket Companies (Views expressed here are my own)
- Contact: lmarkum@live.com, https://www.linkedin.com/in/leemarkum/ , @leemarkum on X

Setting Expectations

- The goal is to introduce you to useful modern features, not provide in-depth coverage. That would take several days.
- I am only considering features from the database engine and not SSIS, SSRS, or SSAS.
- There are a lot of features I could have chosen from.

Modern SQL Server Features That Make Life Better

- How did I decide which SQL Server versions to discuss?
- https://sqlserverupdates.com/
- Extended support for SQL Server 2012 ended in July 2022.
- Extended Support for SQL Server 2014 ended in July 2024.

Agenda

Discuss features in the following categories:

- Performance
- Troubleshooting
- T-SQL
- High Availability/Disaster Recovery

Modern SQL Server Features That Make Life Better

Performance Features

Performance Features

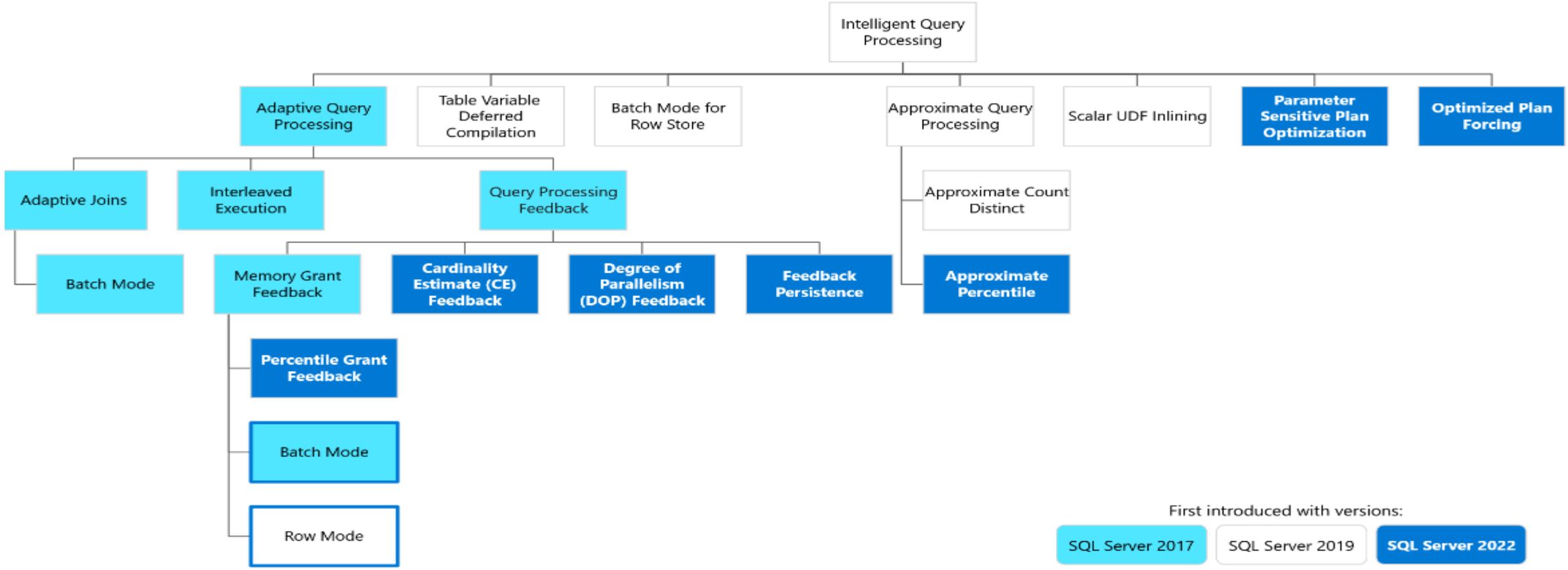
- Lightweight Query Profiling

- https://bit.ly/3N2drhB (Query Profiling Infrastructure - SQL Server | MS Learn)

    - sys.dm_exec_query_profiles (SQL 2014 SP2 but enhanced in later versions)
    - Use Trace Flag 7412 to enable globally in SQL Server 2016SP1 and SQL Server 2017.
    - On by default in SQL Server 2019.
    - Can be used to see index creation progress, which wasn't really possible prior

Performance Features

Intelligent Query Processing Picture from MS Learn
https://bit.ly/3C0Zn1B



Performance Features

- Memory grant feedback (SQL Server 2017)

- Does not require Query Store as it affects cached plans
- Can reduce spills to disk by increasing memory grants
- Can reduce wasted memory by lowering memory grants that are too high
- Can result in a “zig-zag” effect where memory grants are constantly shifting up and down. In this scenario, memory grant feedback disables itself.

Performance Features

- Memory grant feedback (SQL Server 2017)

  Feedback is lost on SQL Server restart or use of RECOMPILE

  Needs database compatibility level 140 for batch mode and 150 for row mode

  Memory grant feedback activity is visible via the XEvent called memory_grant_updated_by_feedback

Performance Features

- Persisted and Percentile memory grant feedback (SQL Server 2022)

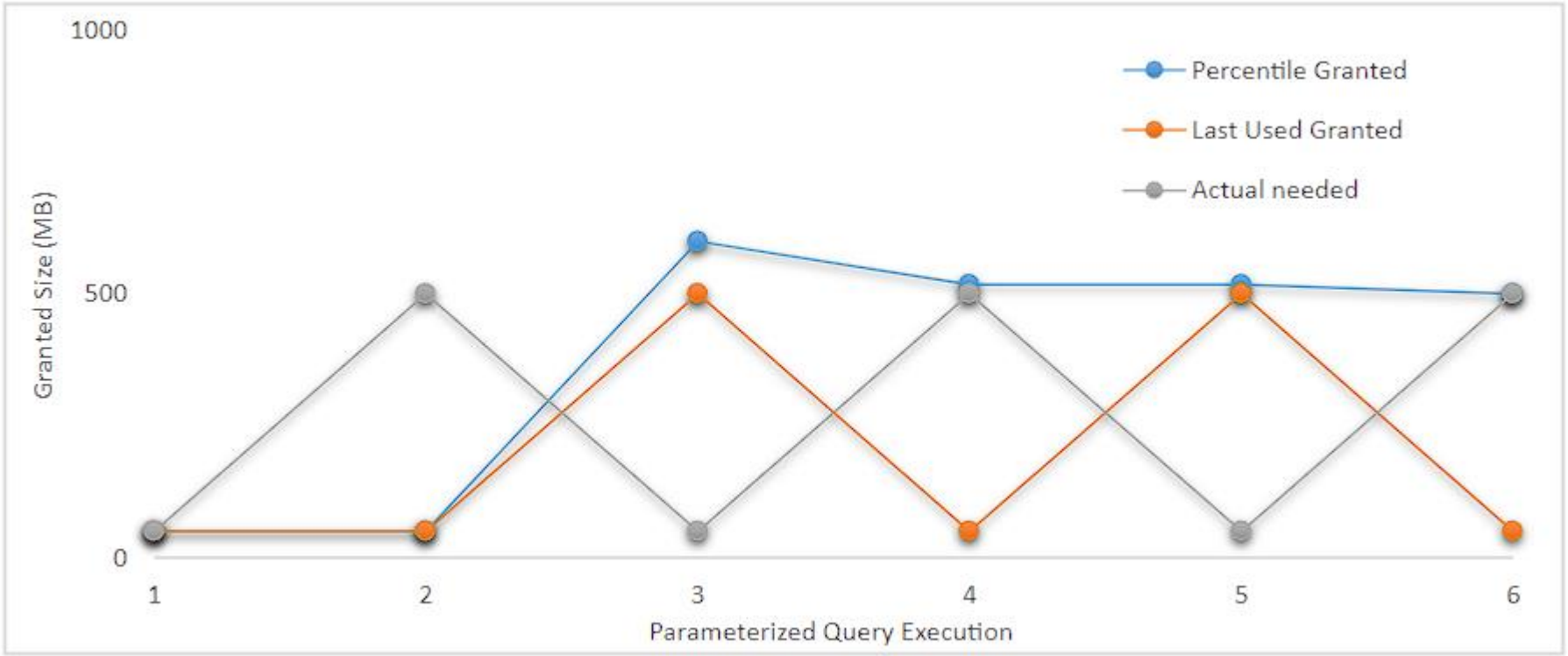
  - Needs compat level 140 and higher.

  - Needs Query Store enabled and in read-write mode.

Performance Features

- Persisted and Percentile memory grant feedback (SQL Server 2022)

https://bit.ly/485EOQZ MS Learn Memory Grant Feedback



Performance Features

- DOP_Feedback (SQL Server 2022)

- Needs compat level 160

- Needs Query Store enabled and in read/write mode

- Only verified feedback is persisted

- Feedback will be persisted in the sys.query_store_plan_feedback catalog view when we reach a stable degree of parallelism feedback value.

Performance Features

Query Store(2016)

- https://bit.ly/2JGUC5J (Lee Markum - Overview of Query Store)
- https://bit.ly/4cUUgjO (Erin Stellato - Why you Need Query Store)
- https://bit.ly/3W61tqw (Erin Stellato - Query Store Best Practices)

- Stores query runtime performance from the perspective of CPU, Duration, logical reads and other metrics
- Wait Stats information is available in SQL 2017 and above
- Custom Capture Policies in 2019 to allow better control over what is captured so Query Store performs better and collects more filtered data.

Performance Features

Query Store(2016)

- Is enabled at the database level
- Stores query plans and related performance metrics
- Allows you to detect regression and force “good query plans”
- Various reports show performance graphically

Performance Features

Query Store(2016)

- Use TF 7745 to prevent Query store data from writing to disk before shutdown (This TF is not available in AWS RDS)
- Use TF 7752 to enable Query Store to load data into memory asynchronously and not prevent query execution while it loads. This is the default in 2019.
- Fantastic to use as a before/after for migrations
- Query Store in SQL Server 2022

Modern SQL Server Features That Make Life Better

Troubleshooting Features

Troubleshooting Features

Verbose Truncation Warnings(2019)

```
Msg 8152, Level 16, State 30, Line 13
String or binary data would be truncated.
The statement has been terminated.
CREATE TABLE DemoSQL2019
(
 [ID] INT identity(1,1),
 [NAME] VARCHAR(10),
)
INSERT INTO DemoSQL2019 VALUES ('SQLSaturday Columbus, OH')
INSERT INTO DemoSQL2019  VALUES ('Brent Ozar Author')
```

Troubleshooting Features

Verbose Truncation Warnings(2019)

Msg 2628, Level 16, State 1, Line 13

String or binary data would be truncated in table 'TruncateMessageDemo.dbo.DemoSQL2019', column 'NAME'.

Truncated value: 'SQLSaturda'.

- Can be turned on in SQL Server 2017 using trace flag 460.
- Available in compatibility level 150 (SQL Server 2019) by default

Troubleshooting Features

Temporal Tables/System-Versioned Tables (2016)

- Composed of a “normal” table and a history table
- Special time range start and time range end columns are added to the normal and history table.
- A temporal table must have a Primary Key

Troubleshooting Features

Temporal Tables/System-Versioned Tables (2016)

- Can be a replacement for triggers as a means of tracking changes to tables.
- Temporal Tables do not track WHO made the modification. But see Aaron Bertrand's solution for that here on MSSQLTips: https://bit.ly/3672sNH
- For tracking WHO as well as WHAT, see the new Ledger feature (SQL 2022)
- Can help you answer, what happened to the data and when, making the feature useful for auditing

Troubleshooting Features

Temporal Tables/System-Versioned Tables (2016)

- Run queries to show trends over time
- Oops deletes and temporal tables.
- No need for side by side restore of a database to examine and recover data, saving hours of delays to start fixing data on a VLDB.
- Use the StartTime and EndTime values for rows to find the records you need to restore from the history table.

- https://bit.ly/4fFQyNa - MSSQLTips.com example
- https://bit.ly/3WDgyAf - Bert Wagner example

Troubleshooting Features

Temporal Tables/System-Versioned Tables (2016)

- An example: Increased the product price by 10% two times.

SysStartTime	SysEndTime	ListPrice	ProductID	Name
2024-01-18 12:48:52.3825924	2024-01-18 12:50:22.7101399	549.83	822	ML Road Frame-W - Yellow, 38
2024-01-18 12:48:52.3825924	2024-01-18 12:50:22.7101399	549.83	833	ML Road Frame-W - Yellow, 40
2024-01-18 12:48:52.3825924	2024-01-18 12:50:22.7101399	549.83	834	ML Road Frame-W - Yellow, 42
2024-01-18 12:48:52.3825924	2024-01-18 12:50:22.7101399	549.83	835	ML Road Frame-W - Yellow, 44
2024-01-18 12:48:52.3825924	2024-01-18 12:50:22.7101399	549.83	836	ML Road Frame-W - Yellow, 48
2024-01-18 12:50:22.7101399	2024-01-18 12:51:09.2742484	604.813	822	ML Road Frame-W - Yellow, 38
2024-01-18 12:50:22.7101399	2024-01-18 12:51:09.2742484	604.813	833	ML Road Frame-W - Yellow, 40
2024-01-18 12:50:22.7101399	2024-01-18 12:51:09.2742484	604.813	834	ML Road Frame-W - Yellow, 42
2024-01-18 12:50:22.7101399	2024-01-18 12:51:09.2742484	604.813	835	ML Road Frame-W - Yellow, 44
2024-01-18 12:50:22.7101399	2024-01-18 12:51:09.2742484	604.813	836	ML Road Frame-W - Yellow, 48

Modern SQL Server Features That Make Life Better

T-SQL Enhancements Features

T-SQL Features

- FileTable(SQL Server 2012) – Could use this to hold and search pictures from a table (photography hobby or business inventory on a sales page) https://bit.ly/3YAOGPO Simple-talk Louis Davidson

- OFFSET FETCH for pagination(2012) Shows ‘x’ number of rows on a page https://bit.ly/3YK1nGH Aaron Bertrand sqlperformance

- THROW for raising errors (2012) – condenses the syntax needed to raise an error and replaces RAISERROR in many cases

T-SQL Features

- DATEFROMPARTS (SQL Server 2012)
- TIMEFROMPARTS (SQL Server 2012)
- DATETIMEFROMPARTS (SQL Server 2012)
- EOMONTH (SQL Server 2012) – Prevents the need for writing a UDF

T-SQL Features

- Inline specification of CLUSTERED and NONCLUSTERED indexes (2014)
- DROP IF EXISTS(SQL Server 2016SP1)
- CREATE OR ALTER(2016SP1) - Can be important for Query Store

T-SQL Features

- JSON format support(2016) but enhancements in 17, 22 and Azure SQL DB.
- https://bit.ly/46MF6vm Hasan Savran - New Features in SQL Server for Developers
- ALTER TABLE WITH ONLINE = ON (2016)
- New string functions (2017) CONCAT_WS, TRANSLATE, and TRIM, and WITHIN GROUP is now supported for the STRING_AGG function

T-SQL Features

Inline Index Creation:

```
USE DBAUtility;
GO
DROP TABLE IF Exists t1;

--filtered index
CREATE TABLE t1
(
        c1 INT,
        index IX1 (c1) WHERE c1 > 0
);
```

T-SQL Features

Inline Index Creation:

```
DROP TABLE IF Exists t2;

--multi-column index
CREATE TABLE t2
(
        c1 INT,
        c2 INT,
        INDEX ix_1 NONCLUSTERED (c1,c2)
);
```

T-SQL Features

Inline Index Creation:

```
DROP TABLE IF Exists t3;

--multi-column unique index
CREATE TABLE t3
(
        c1 INT,
        c2 INT,
        INDEX ix_1 UNIQUE NONCLUSTERED (c1,c2)
);
```

T-SQL Features

Inline Index Creation:

```
DROP TABLE IF Exists t4;

-- make unique clustered index
CREATE TABLE t4
(
        c1 INT,
        c2 INT,
        INDEX ix_1 UNIQUE CLUSTERED (c1,c2)
);
```

T-SQL Features

Traditional Concatenation, CONCAT and CONCAT_WS

```
USE WideWorldImporters;
GO
SELECT          (DeliveryAddressLine2+' '+ C.CityName+' '+DeliveryPostalCode) AS OlderConcatMethod,
CONCAT          (DeliveryAddressLine2,' ',C.CityName,' ',DeliveryPostalCode) AS NewConcatMethod,
CONCAT_WS       (' ',DeliveryAddressLine2,C.CityName,DeliveryPostalCode) AS CONCAT_WSMethod
FROM [WideWorldImporters].[Sales].[Customers] AS CUST
INNER JOIN [Application].[Cities] AS C ON CUST.DeliveryCityID = C.CityID;
```

T-SQL Features

Traditional Concatenation, CONCAT and CONCAT_WS

Results | Messages

	OlderConcatMethod	NewConcatMethod	CONCAT_WSMethod
1	516 Bojin Road Abbottsburg 90692	516 Bojin Road Abbottsburg 90692	516 Bojin Road Abbottsburg 90692
2	1161 Chang Lane Absecon 90540	1161 Chang Lane Absecon 90540	1161 Chang Lane Absecon 90540
3	1268 Gonzales Road Accomac 90378	1268 Gonzales Road Accomac 90378	1268 Gonzales Road Accomac 90378
4	328 Bhat Street Aceitunas 90213	328 Bhat Street Aceitunas 90213	328 Bhat Street Aceitunas 90213
5	232 Abbasi Avenue Airport Drive 90211	232 Abbasi Avenue Airport Drive 90211	232 Abbasi Avenue Airport Drive 90211
6	1280 Tran Boulevard Akhiok 90197	1280 Tran Boulevard Akhiok 90197	1280 Tran Boulevard Akhiok 90197
7	1278 Lam Street Akhiok 90112	1278 Lam Street Akhiok 90112	1278 Lam Street Akhiok 90112

T-SQL Features

SQL Server 2022 Enhancements:

- Greatest/Least (eliminates the need for a complex searched CASE or a CROSS APPLY)
- GENERATE_SERIES
- Window or STRING_SPLIT with the enable_ordinal enhancement enabled
- Create index and resumable constraint enhancements

T-SQL Features

GREATEST/LEAST:

A platform you support allows customers to manage email subscriptions to a newsletter. The customer wants to know in which month was the largest and smallest number of opened newsletters from the emails .

https://bit.ly/3M38Dao Examples taken from a SQL Server 2022 article by Aaron Betrand

T-SQL Features

GREATEST/LEAST:

Your summarized data table looks like this:

```
CREATE TABLE dbo.SummarizedEmailOpens
(
Year int,
Jan  int,
Feb  int,
Mar  int,
Apr  int,
May  int,
Jun  int,
Jul  int,
Aug  int,
Sep  int,
Oct  int,
Nov  int,
[Dec] int
);
```

T-SQL Features

GREATEST/LEAST:

You insert some rows like this:

```
INSERT dbo.SummarizedEmailOpens
VALUES
(2021, 55000,81000,74000,34598,98765,67821,56903,12586,40861,12345,78944,50349),

(2022, 55000,521904,74000,765401,103548,200000,135700,12586,40861,12345,78944,50349);
```

T-SQL Features

GREATEST/LEAST:

CASE Statement to compare just 3 months:

```
SELECT Year,
  BestMonth = CASE
    WHEN Jan > Feb THEN
      CASE WHEN Jan > Mar THEN Jan ELSE Mar END
    ELSE
      CASE WHEN Mar > Feb THEN Mar ELSE Feb END
    END,
  WorstMonth = CASE
    WHEN Jan < Feb THEN
      CASE WHEN Jan < Mar THEN Jan ELSE Mar END
    ELSE
      CASE WHEN Mar < Feb THEN Mar ELSE Feb END
    END
FROM dbo.SummarizedEmailOpens;
```

T-SQL Features

GREATEST/LEAST:

CROSS APPLY to compare just 3 months:

```
SELECT Year,
  BestMonth  = MAX(MonthlyTotal),
  WorstMonth = MIN(MonthlyTotal)
FROM
(
  SELECT s.Year, Months.MonthlyTotal
  FROM dbo.SummarizedEmailOpens AS s
  CROSS APPLY (VALUES([Jan]),([Feb]),([Mar])) AS
[Months](MonthlyTotal)
) AS Sales
GROUP BY Year;
```

T-SQL Features

Solution with GREATEST/LEAST:

```
SELECT Year,
  BestMonth  = GREATEST([Jan],[Feb],[Mar]),
  WorstMonth = LEAST   ([Jan],[Feb],[Mar])
FROM dbo.SummarizedEmailOpens;
```

Modern SQL Server Features That Make Life Better

High Availability/Disaster Recovery

High Availability/Disaster Recovery

Log Shipping -

The readable secondary in standby means data is not up to date.

No single built-in connection name

Config is for single databases

Database mirroring -

Secondaries are not readable

Work-around is database snapshots

config is for single databases

High Availability/Disaster Recovery

Replication -

- Lots of moving parts - you love Agent Jobs, right?!
- Need a PK on each replicated table
- No single built-in connection name for the application
- Snapshots to get data seeded can be resource intensive

Failover Clustering -

- Shared storage or shared storage trickery required.
- Only provides HA

High Availability/Disaster Recovery

Availability Groups - (2012) but many enhancements all the way through 2022

How do availability groups try to address the limitations we just discussed?

High Availability/Disaster Recovery

Summary of Availability Group Positives

- No shared storage or shared storage “trickery” via software (solves FCI obstacle)
- Provides built-in DR via multi-subnet config (solves FCI obstacle)
- Tables don’t need a PK to participate (solves replication obstacle)

High Availability/Disaster Recovery

Summary of Availability Group Positives

- Built-in connection name for failover called the Listener
  (solves LS/replication obstacle)
- Data is within a few seconds of the primary. (Solves LS obstacle)
- Readable secondaries by direct connect to a secondary instance name (Solves Mirroring/LS obstacle)

High Availability/Disaster Recovery

Summary of Availability Group Positives

- Readable secondary using ReadOnly routing feature via Listener
  (Solves Mirroring/LS obstacle)

- Automatic failover for a group of databases (Solves Mirroring/LS/replication obstacle)

High Availability/Disaster Recovery

Summary of Availability Group Positives

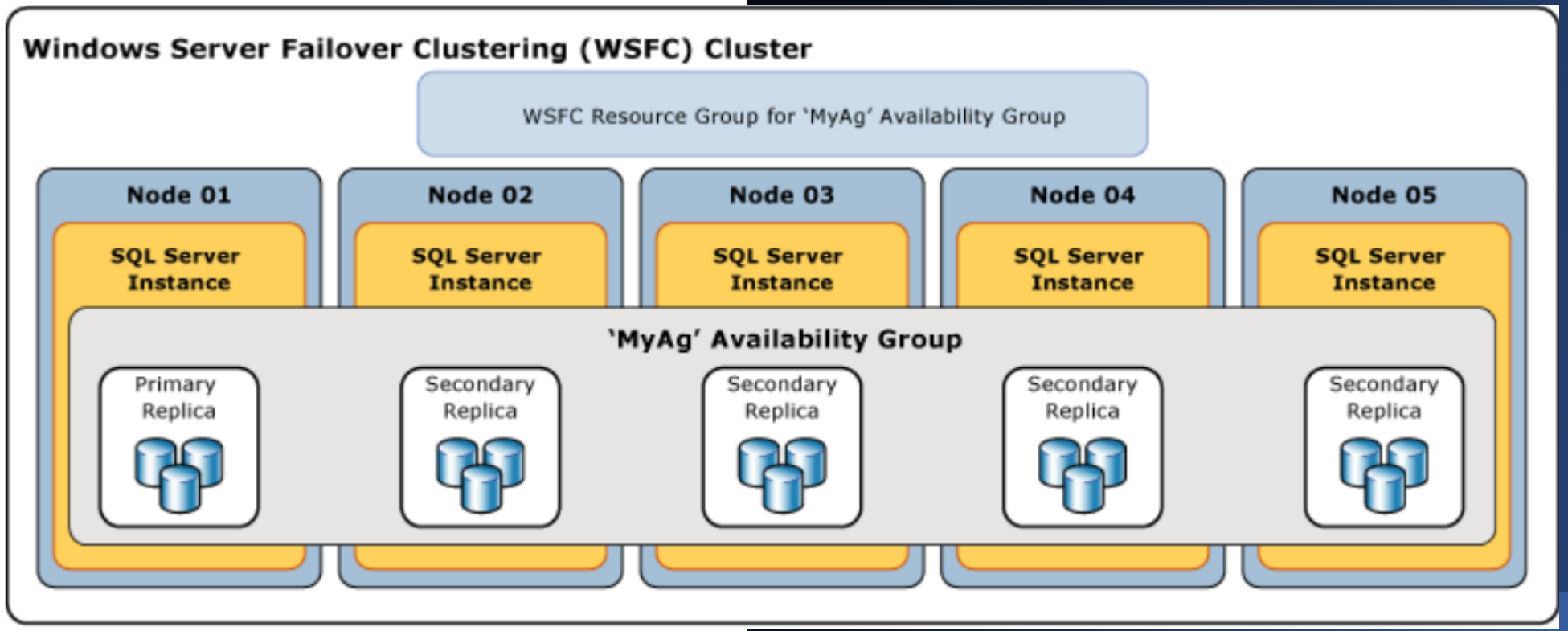
- Relatively easy to leverage for upgrades
- Automatic page repair from a secondary under certain conditions (Trackable with sys.dm_hadr_auto_page_repair)
- Manage objects outside the database by contained AGs(SQL 2022)
- Query Store on readable secondaries (SQL 2022)

Modern SQL Server Features That Make Life Better

HA/DR Category

Availability Groups (2012 and Up, but don't try it on 2012)
https://bit.ly/3xieeks (MS Docs - What is an AlwaysOn Availability Group)
https://bit.ly/3dqxjrG (Lee Markum - My Availability Group Database Isn't Synchronizing)

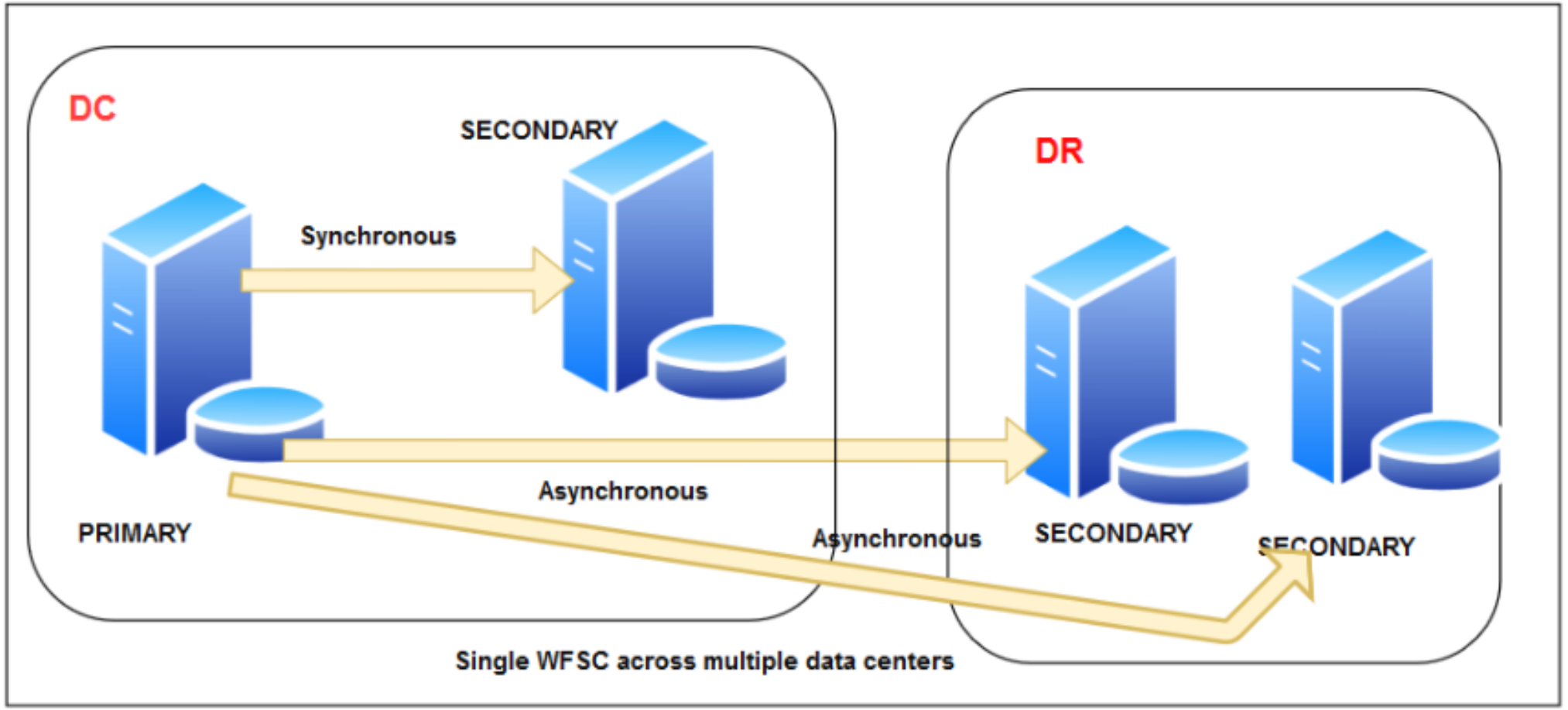
https://bit.ly/3xieeks
MS Docs depiction



High Availability/Disaster Recovery

Primary Data Center

Disaster Recovery Location



Multi-subnet Availability Group Depiction from SQLShack: https://www.sqlshack.com/sql-server-always-on-listeners/

High Availability/Disaster Recovery

Things You Need to Make AGs Successful

- The proper team members
  - Multiple people who understanding Windows Clustering well
  - Multiple DBAs who understand the AG components well
  - Multiple network SMEs
- Knowledge of cloud provider requirements for AGs in IaaS

High Availability/Disaster Recovery

Multi-subnet in AWS IaaS

https://bit.ly/3YwYSZi AWS Documentation

- Requires the AG to be multi-subnet.
- Strongly suggested to use multiple Availability Zones.
- EVERY ec2 instance in the AG has to be in its own subnet.
- Storage can be ebs or FSx Netapp for ONTAP.
- A story about upgrading SQL Server AGs in AWs IaaS

High Availability/Disaster Recovery

Accelerated Database Recovery(2019)

https://www.mssqltips.com/sqlservertip/5971/accelerated-database-recovery-in-sql-server-2019/

1. Uses the “persisted version store” in the user DB to do a logical revert
2. Processes the TLOG from the last checkpoint
3. Off by default in on-prem SQL Server
4. On by default in SQL Database and Managed Instance and can’t be disabled
5. Multi-threaded version cleanup available in 2022 up to the # of cores with the ADR Cleaner Thread Count option in sp_configure

High Availability/Disaster Recovery

Accelerated Database Recovery(2019)

	BusinessEntityID	EmailAddressID	EmailAddress	rowguid	ModifiedDate
1	1	1	ken0@adventure-works.com	8A1901E4-671B-431A-871C-EADB2942E9EE	2009-01-07 00:00:00.000
2	2	2	terri0@adventure-works.com	B5FF9EFD-72A2-4F87-830B-F338FDD4D162	2008-01-24 00:00:00.000
3	3	3	roberto0@adventure-works.com	C8A51084-1C03-4C58-A8B3-55854AE7C499	2007-11-04 00:00:00.000
4	4	4	rob0@adventure-works.com	17703ED1-0031-4B4A-AFD2-77487A556B3B	2007-11-28 00:00:00.000
5	5	5	gail0@adventure-works.com	E76D2EA3-08E5-409C-BBE2-5DD1CDF89A3B	2007-12-30 00:00:00.000
6	6	6	jossef0@adventure-works.com	A9C4093A-4F4A-4CAD-BBB4-2C4E920BACCB	2013-12-16 00:00:00.000
7	7	7	dylan0@adventure-works.com	70429DE4-C3BF-4F19-A00A-E976C8017FB3	2009-02-01 00:00:00.000
8	8	8	diane1@adventure-works.com	37F02A87-058D-49F8-A20D-965738B0A71F	2008-12-22 00:00:00.000
9	9	9	gigi0@adventure-works.com	F888A16D-0C33-459E-9D72-D16AE0BB1F43	2009-01-09 00:00:00.000
10	10	10	michael6@adventure-works.com	E0DD366D-433D-4F5A-9347-1A5FE7FBE0A3	2009-04-26 00:00:00.000

Modern SQL Server Features That Make Life Better

How can you discover information about new SQL Server features?

- https://docs.microsoft.com/en-us/sql/sql-server/what-s-new-in-sql-server-2016?view=sql-server-ver15 (The page has links in the left window of the site to 2016-2019)
- https://docs.microsoft.com/en-us/previous-versions/sql/ (You guessed it, everything prior to SQL 2016)
- Various MSSQLTIPS articles: https://www.mssqltips.com/sqlservertip/4574/new-features-in-sql-server-2016-service-pack-1
- https://www.mssqltips.com/sqlservertip/5376/tsql-enhancements-in-sql-server-2017/
- https://blog.pythian.com/top-10-new-features-of-sql-server-2019/
- https://www.sentryone.com/blog/aaronbertrand/fishing-for-features-in-ctps
- https://www.sentryone.com/blog/aaronbertrand/more-changes-sql-server

Modern SQL Server Features That Make Life Better

Bonus Material

- SSISDB catalog (SQL 2012)
- In-Memory OLTP(SQL 2014) – This can produce significant improvements for some workloads.
- Backup encryption (2014)
- Always Encrypted (2016)
- Dynamic Data Masking(2016)
- Row Level Security(2016)
- ColumnStore Indexes (2012, but many enhancements since)
- Data Virtualization/External tables (2019)
- Forced Plan Optimization (2022)
- CE_Feedback (2022)
- Tempdb metadata optimization (2022)

Modern SQL Server Features That Make Life Better

Summary

- Discussed features in the following categories:
  - Performance
    - Intelligent Query Processing, Query Store
  - Troubleshooting
    - Verbose truncation warnings, temporal tables
  - T-SQL
    - OFFSET FETCH, THROW, new date and datetime related functions
  - High Availability/Disaster Recovery
    - Availability Groups, Accelerated Database Recovery

Modern SQL Server Features That Make Life Better

Thank you for attending!

Modern SQL Server Features That Make Life Better

QUESTIONS?

Contact Information

- Twitter(X): @leemarkum
- LinkedIn: https://www.linkedin.com/in/leemarkum/
- Email: lmarkum@live.com