

The SQL Server Extended Development Environment



George Walkey

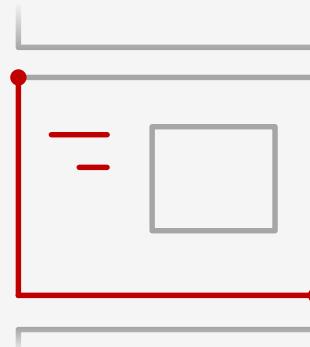
Patient First Corporation

<https://github.com/gwalkey>

george.walkey@patientfirst.com

Your life of (Data) Crime

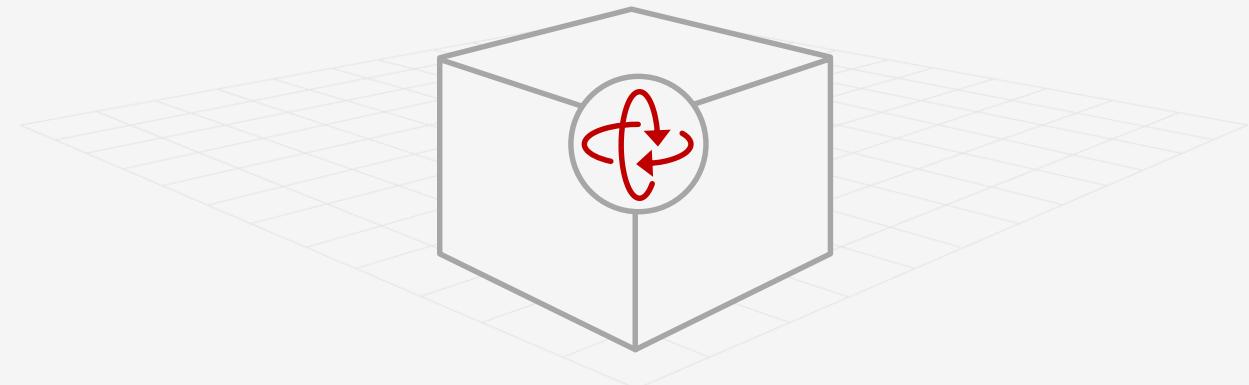
Data Developer



Does things with Data

- Data Architect
- Data Engineer
- Data Scientist

Database Management (DBA)



Keeps the place running

- Alerts
- Dashboards
- Instrumentation

A Day in the Life

Typical Day

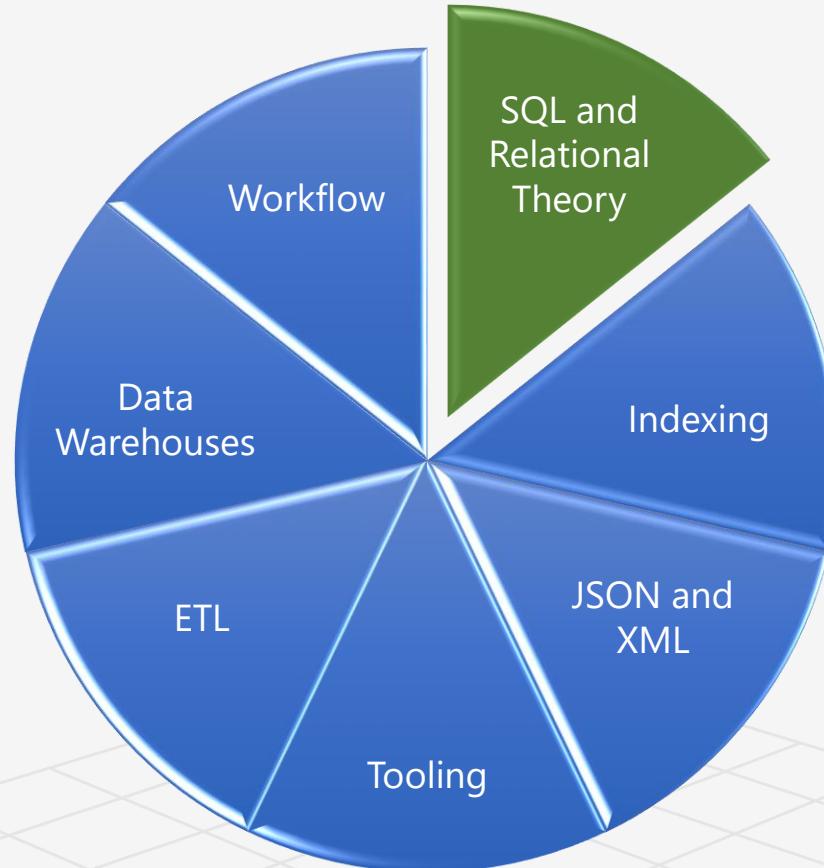
- Get Requirements from Project team
- Coffee
- Review Failed Overnight Jobs
- Run Daily Manual Morning Processes
- Check Scrum Board for Work In Process
- Make TODO List of pending Scrum PBI you solved last night in the shower
- Attend Scrum Stand up
- Ask boss to talk to MainFrame Team about X
- Get info from Project Manager
- Give info to Project Manager
- Coffee
- Fix Bad User State in Onboarding App DB - add to ToDo List
- Research using Powershell to download FTP files
- Create Wiki on a process you do/fix regularly and want to delegate
- Upgrade 2 SQL Servers from 2017-2019
- Respond to BAs Active Directory request
- Lunch
- Update Power BI Dashboard for Open PBI - Add New Patient Volume Tab
- Help Systems Team NOT mess up my backup chains with Veeam
- Do more analysis on Claim Rejections data - How to Group SmartEdit Errors?
- Read Daily Blog Roll - Kevin Feasel's Curated SQL et al
- Check Notepad++ for updates
- Coffee
- Check Weekend Weather And Traffic
- ETC ETC ETC

A Day in the Life

A Typical Project

- Get Requirements from Project team
- Find Data
- Ask if there is a way to automate the download
- Get the Data
- Evaluate the Data - extract schema
- Build a Load Table
- Build an ETL
- Load Data
- Clean Data
- Evaluate typical SQL queries and add appropriate Indexes
- Create relational tables if needed, PK/FK
- Add Column constraints
- Build SSRS or Power BI
- Evaluate REAL SQL sent to DB from SQL clients and create REAL Indexes this time
- Automate ETL Process and create Power BI dataset refresh schedule
- Check in the ETL Code
- Check in the SSRS/Power BI
- Create Database Project in Visual Studio from Prod DB and check in
- Deploy bits from Dev to Prod
- Grant Permissions to SSRS/Power BI to Product Owners
- Write Wiki
- Add Error handling to ETL
- Update Wiki

Data Development



Data Development

1

SQL and Relational Theory

- SQL is still our Bread and Butter – SET-Based thinking
- CRUD:
 - Create - INSERT
 - Read - SELECT
 - Update - UPDATE
 - Delete - DELETE
- Stored Procedures, Views, Triggers, Functions
- 20,000 line Stored Procedures are fun to Debug*
- Declarative Referential Integrity (DRI) - Delete Cascade/Restrict
- Data Integrity - Column Domains/Constraints
- Left Joins, CTEs, SubQueries, Windowing functions

Data Development

2

Judicious Indexing

- Suggested SSMS "Green" Indexes are not always helpful
- SSMS does not always suggest Indexes that ARE helpful
- SSMS will suggest an index after suggesting another one first
- Delete indexes that are only being updated – But how long do you wait to tell?
- Create Indexes on your Foreign Keys.
- Keys vs Includes
 - Keys used in WHERE, GROUP BY, ORDER BY and JOIN
 - Includes - just along for the ride
- Columnstore Indexes
 - Used in DataWarehouses, HTAP Databases
 - Are the Primary index type in Power BI/Azure Synapse/Azure Data Explorer
- NOOB Mistake: Create an index on every Column

Data Development

3

JSON and XML

- OpenJSON function, XML XPATH
- XML Indexes - Compression in 2022 - shredding XML shreds CPU

```
SELECT
    Guarantor.*,
    COALESCE(CONVERT(INT, SUBSTRING(GuarantorID, 1, CHARINDEX('.', GuarantorID) - 1)), 0)
FROM
OPENJSON(@jsontext)
WITH
(
    GUHeader NVARCHAR(MAX) '$.Guarantors' AS JSON
) AS GUroot
OUTER APPLY OPENJSON(GUroot.GUHeader)
    WITH (
        GuarantorID VARCHAR(50) '$.Guarantor.GuarantorID',
        FirstName VARCHAR(50) '$.Guarantor.FirstName',
        MiddleInitial VARCHAR(5) '$.Guarantor.MiddleInitial',
        LastName VARCHAR(50) '$.Guarantor.LastName',
        StreetAddress VARCHAR(50) '$.Guarantor.StreetAddress',
        City VARCHAR(50) '$.Guarantor.City',
        State VARCHAR(50) '$.Guarantor.State',
        ZipCode VARCHAR(50) '$.Guarantor.ZipCode',
        CellPhone VARCHAR(15) '$.Guarantor.PhoneNumbers.CellPhone',
        HomePhone VARCHAR(15) '$.Guarantor.PhoneNumbers.HomePhone',
        WorkPhone VARCHAR(15) '$.Guarantor.PhoneNumbers.WorkPhone',
        SocialSecurityNumber VARCHAR(50) '$.Guarantor.SocialSecurityNumber',
        GuarantorEmployer VARCHAR(50) '$.Guarantor.GuarantorEmployer',
        County VARCHAR(50) '$.Guarantor.County',
        AmountDue VARCHAR(15) '$.Guarantor.AmountDue',
        UnbilledCharges VARCHAR(15) '$.Guarantor.UnbilledCharges',
        Payments VARCHAR(15) '$.Guarantor.Payments',
        LastStatementDate VARCHAR(15) '$.Guarantor.LastStatementDate',
        YearToDateCharges VARCHAR(15) '$.Guarantor.YearToDateCharges',
        YearToDatePayments VARCHAR(15) '$.Guarantor.YearToDatePayments',
        LastPaymentDate VARCHAR(15) '$.Guarantor.LastPaymentDate',
        BDWO_PrincipalDue VARCHAR(15) '$.Guarantor.BadDebtWriteOff.PrincipalDue',
        BDWO_FeeDue VARCHAR(15) '$.Guarantor.BadDebtWriteOff.FeeDue',
        BDWO_InterestDue VARCHAR(15) '$.Guarantor.BadDebtWriteOff.InterestDue',
        BDWO_LastUpdate VARCHAR(15) '$.Guarantor.BadDebtWriteOff.LastUpdate',
        IndustrialClientName VARCHAR(15) '$.Guarantor.IndustrialClientName',
        NoStatementReason VARCHAR(15) '$.Guarantor.NoStatementReason',
        AdditionalInformation VARCHAR(15) '$.Guarantor.AdditionalInformation',
        Inactive VARCHAR(15) '$.Guarantor.Inactive',
        GuarantorComments NVARCHAR(MAX) '$.Guarantor.GuarantorComments' AS JSON
    ) AS Guarantor
```

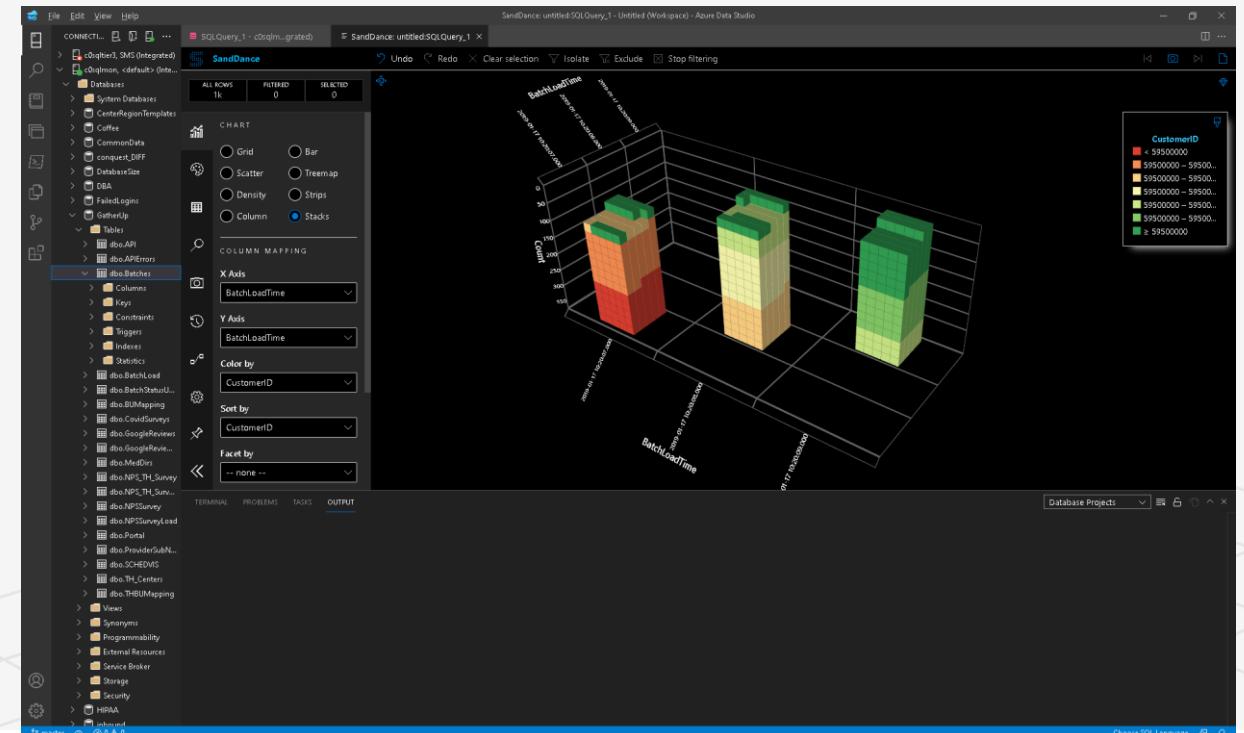
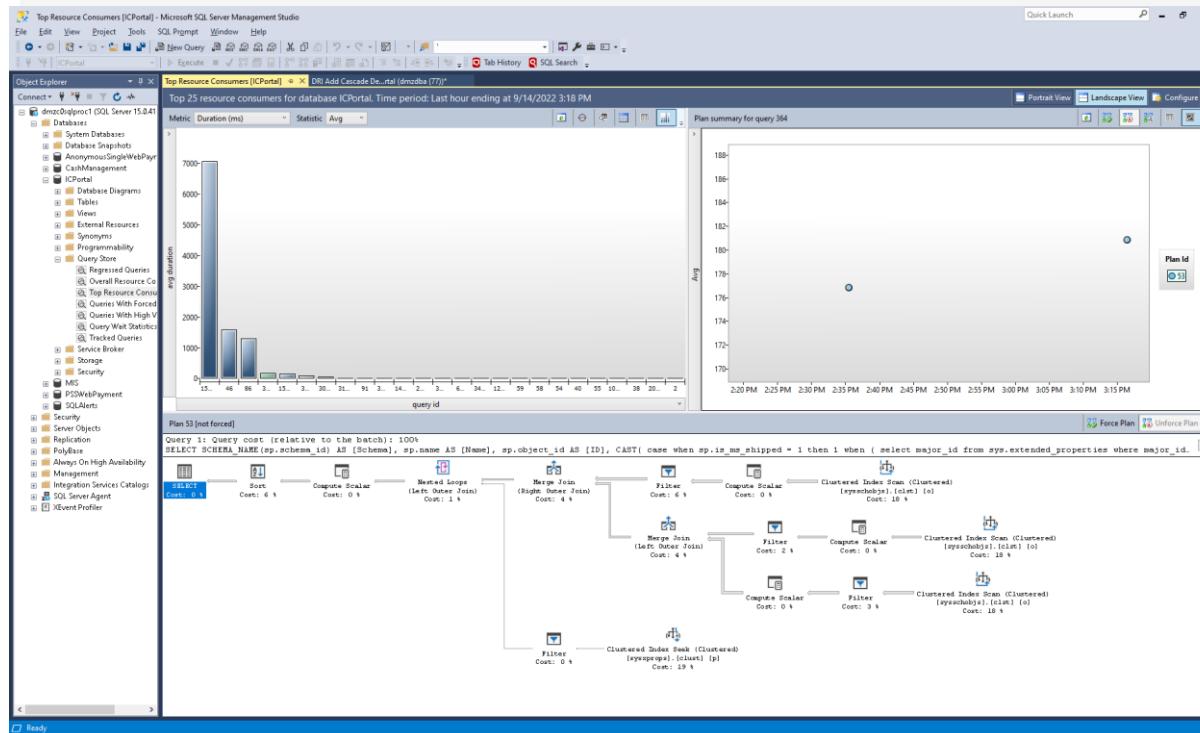
```
bs.b.value('(ns:Field39/text())[1]', 'varchar(50)') AS 'DCN',
CASE
    WHEN
        bs.b.value('(ns:Field34/text())[1]', 'varchar(50)')='Sub Id:'
    THEN
        bs.b.value('(ns:Field35/text())[1]', 'varchar(50)')
    WHEN
        bs.b.value('(ns:Field42/text())[1]', 'varchar(50)')='Sub Id:'
    THEN
        bs.b.value('(ns:Field43/text())[1]', 'varchar(50)')
    WHEN
        bs.b.value('(ns:Field38/text())[1]', 'varchar(50)')='Sub Id:'
    THEN
        bs.b.value('(ns:Field39/text())[1]', 'varchar(50)')
    END AS 'SubID',
bs.b.value('(ns:Field54/text())[1]', 'varchar(512)') AS 'Message',
tc.c.value('(ns:Field50/text())[1]', 'varchar(512)') AS 'Claim Msg Code - Message/Entity Id Codes',
tc.c.value('(ns:Field52/text())[1]', 'varchar(512)') AS 'Claim Msg Code - Message'
FROM
    dbo.RejectionsXML r
CROSS APPLY
    XMLData.nodes('/ns:Report/ns:lstClaimDetails/ns:rgSpecID_Collection/ns:rgSpecID/ns:rgReportType_Collection/ns:rgReportType/ns:rgClaimPageHeader_Collect
OUTER APPLY
    bs.b.nodes('ns:tblClaimMessages/ns:ParentGroup_Collection/ns:ParentGroup/ns:PosSegmentsGroup_Collection[1]/ns:PosSegmentsGroup[1]') AS tc(c)
CROSS APPLY
    r.XMLData.nodes('/ns:Report/ns:lstReportSummary') AS e(e)
WHERE
    ID = @CurrentID
```

Data Development

4

SQL Tooling - SSMS or Azure Data Studio

- SSMS for Most Daily Tasks - 32bit - Control over more Server Objects – Full Mgmt
- Azure Data Studio - 64bit - Notebook Support - Many Extensions – “Viewer”



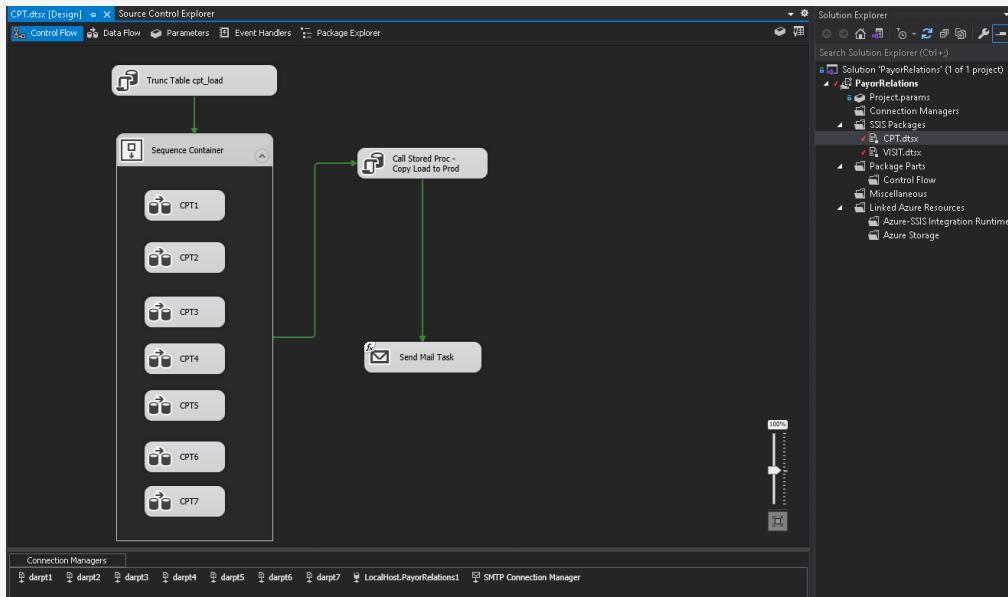
Data Development

4

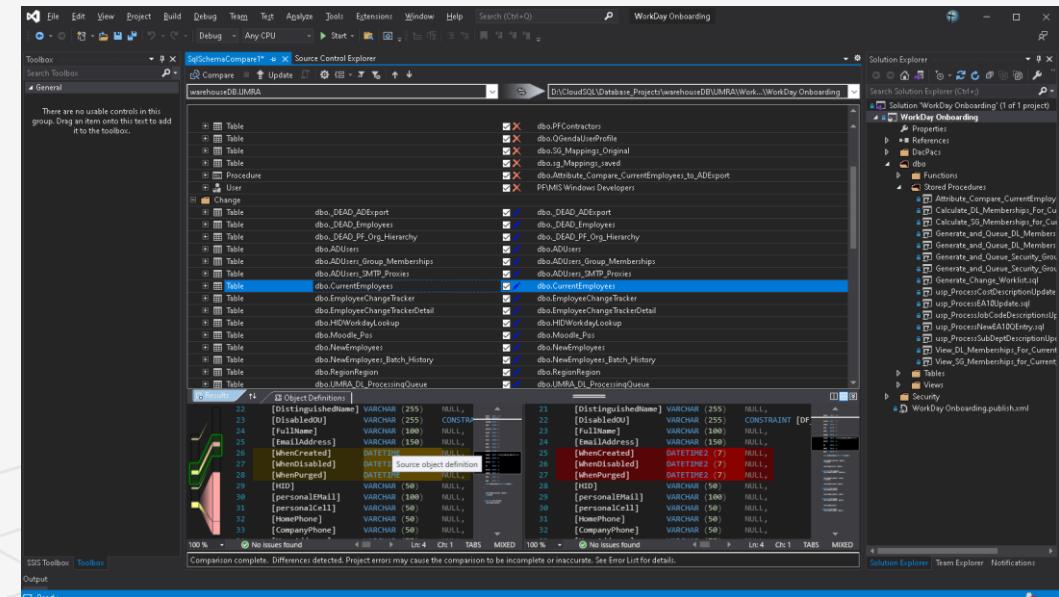
SQL Tooling – Visual Studio

- Create Database, SSIS, SSAS, PowerShell Projects
- With SSDT Extension: Compare Schema and Data
- Free Community Edition - Fully Integrated with Azure DevOps and GitHub

SSIS



Schema Compare

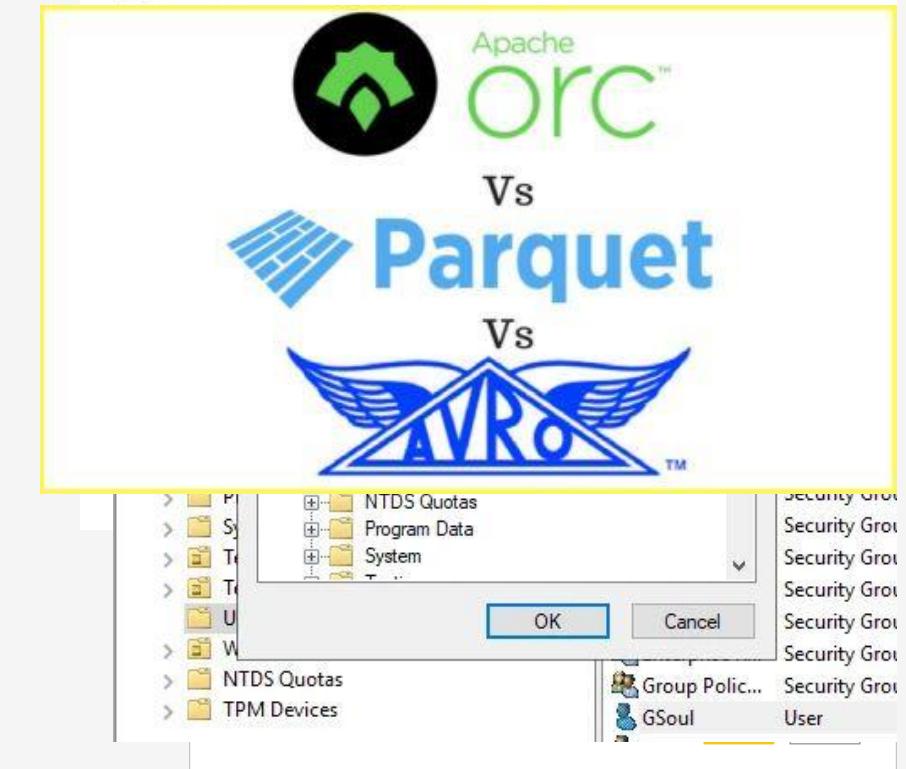


Data Development

5

ETL/ELT - Data Sources

- Non-structured schema-less text files
- XML and JSON
- Exchange Mailboxes with HTML tables
- Rest API paged response payloads
- .NET Streaming SQLReader
- Azure Storage Tables
- Azure Storage Queues
- Active Directory
- Big Data Files
- Power BI:
 - **Use the 100+ ETL transforms then use PBI Dataset as a SQL source**



Data Development

5

ETL/ELT - Tools

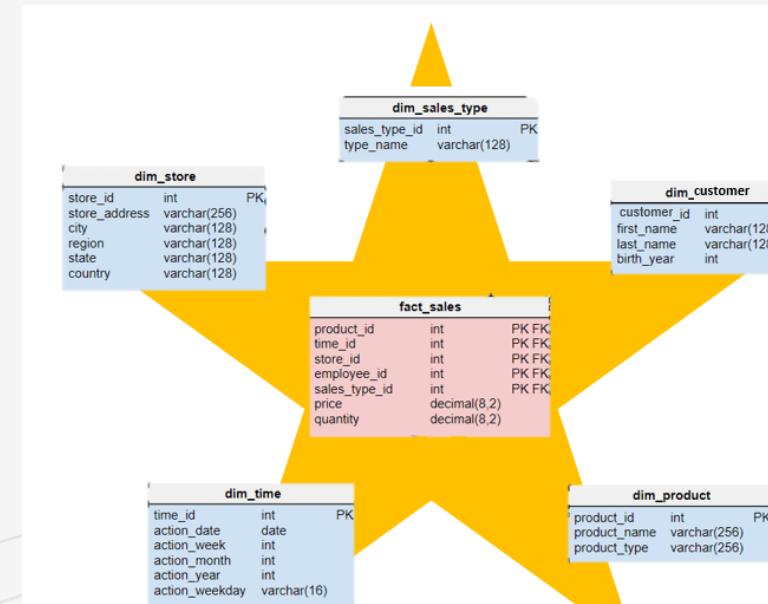
- BulkInsert/BCP - Oldie but goody
- SSIS/ADF
- Powershell 7 - .Net Namespaces for String, XML, JSON object handling
- SQL OpenRowSet, OpenQuery - Big Data in SQL 2022
- R/Python/Java integration

Data Development

6

Data Warehouses

- Are still relevant
- The Star Schema is the basis for all Power BI Models
- Dimensions (Categories) and Facts (Aggregates)
- Practice with AdventureWorksDW Sample Database
- Learn DAX

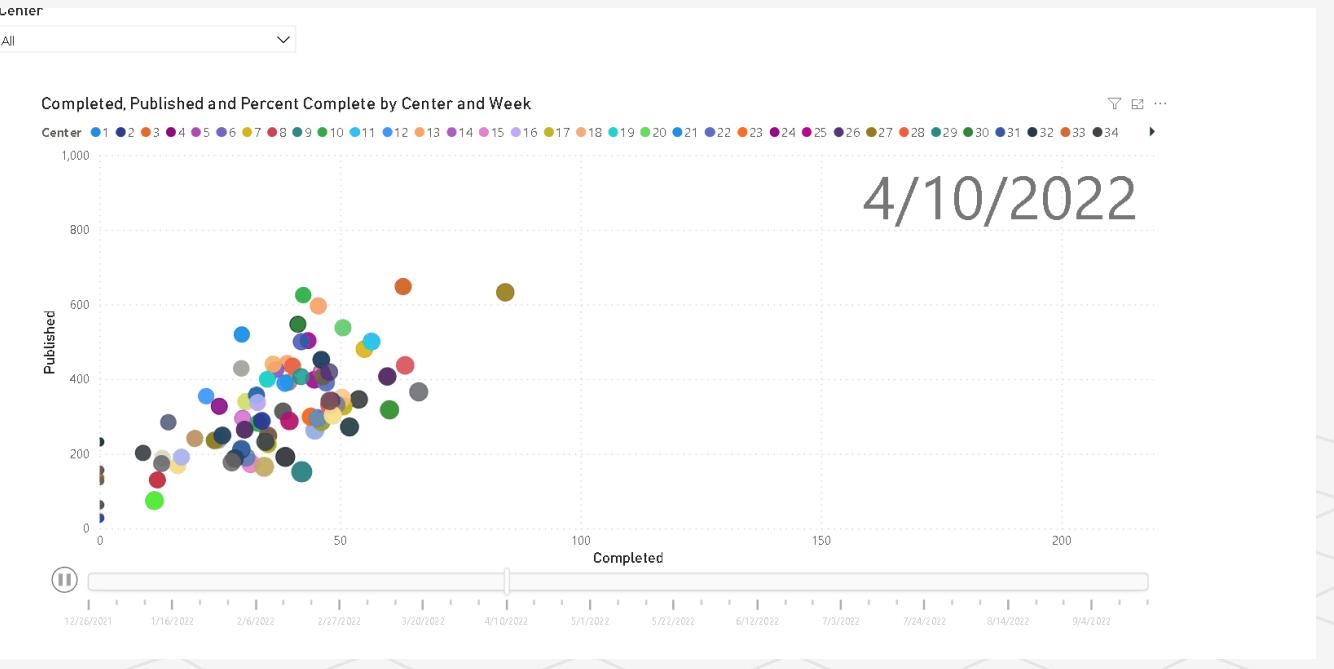
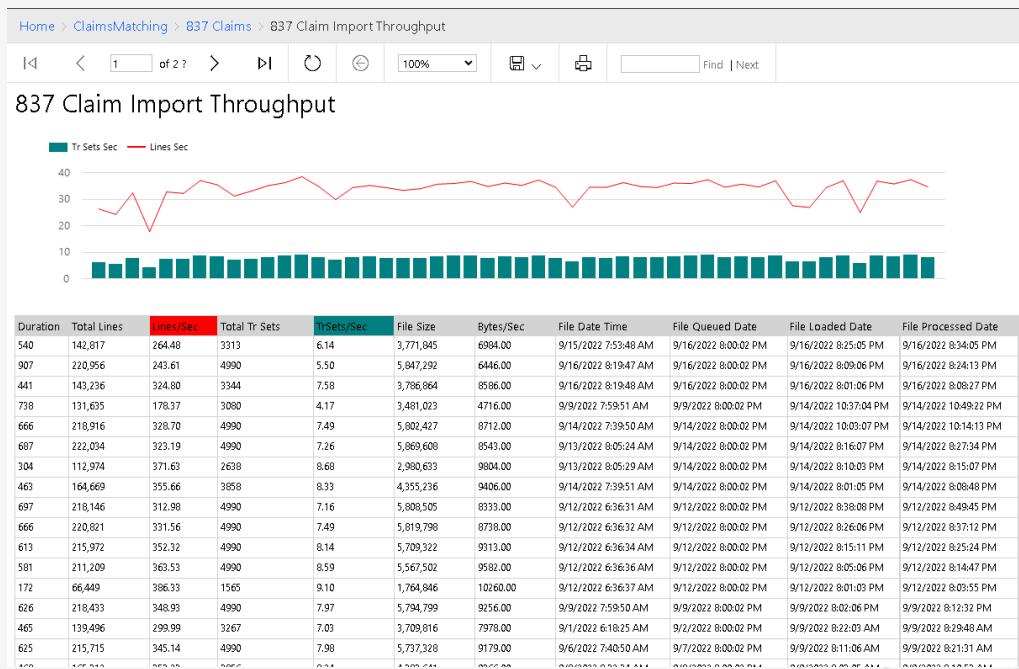


Data Development

7

Reporting

- SSRS is still relevant – FREE - Is in Power BI as Paginated Reports
 - Reports Delivered to mailbox as PDF, Excel, HTML
 - Behind the firewall, no per-user fees, Can run Stored Procedures, Accept Input*
- Power BI / Tableau



Data Development

8

Power BI

- Learn DAX - yes its ~~backwards~~ different, but essential
- Microsoft PL-300 test will kill you, but make you stronger
- Power BI Report Server
 - On-premises Edition
 - Is a Hybrid PBI/SSRS Enterprise edition
 - Includes Data-Driven Subscriptions
- Can Initiate a Power BI Dataset Refresh via REST API
 - Typically last step of an ETL Process

Create and manage aggregations - Learn Microsoft Docs
Data reduction techniques for Import modeling - Power BI Microsoft Docs
Key influencers visualizations tutorial - Power BI Microsoft Docs
Apply AI Insights - Learn Microsoft Docs
Connect to AI Insights in Power BI Desktop - Power BI Microsoft Docs
Use grouping and binning in Power BI Desktop - Power BI Microsoft Docs
DAX - SUMX of IF: A Perfect Blend of Simple & Sophisticated - Power BI Microsoft Docs
Distribute content to external guest users with Azure AD B2B - Power BI Microsoft Docs
Publish an app in Power BI - Power BI Microsoft Docs
Distribute a report or dashboard - Learn Microsoft Docs
Run and view insights on dashboard tiles - Power BI Microsoft Docs
Create dynamic reports with parameters - Learn Microsoft Docs
Parameters - Power Query Microsoft Docs
Filter a report using query string parameters in the URL - Power BI Microsoft Docs
Configure a real-time dashboard - Learn Microsoft Docs
Real-time streaming in Power BI - Power BI Microsoft Docs
Power BI dashboard integration with Azure Stream Analytics - Power BI Microsoft Docs
Use the Analytics pane in Power BI Desktop - Power BI Microsoft Docs
Conduct time series analysis - Learn Microsoft Docs
Use quick measures for common and powerful calculations - Power BI Microsoft Docs
High-density scatter charts in Power BI - Power BI Microsoft Docs
Anomaly detection tutorial - Power BI Microsoft Docs
How To Detect Anomalies And Outliers In Your Data - Microsoft Docs
Bi-directional relationship guidance - Power BI Microsoft Docs
Many-to-many relationship guidance - Power BI Microsoft Docs
Configure data alerts - Learn Microsoft Docs
Dataset connectivity and management with the XMLA endpoint - Power BI Microsoft Docs
Creating a dataflow - Power BI Microsoft Docs
Many-to-many relationships in Power BI Desktop - Power BI Microsoft Docs
Configure incremental refresh settings - Learn Microsoft Docs
Use grouping and binning in Power BI Desktop - Power BI Microsoft Docs
Identify outliers with Power BI visuals - Learn Microsoft Docs
Configure a real-time dashboard - Learn Microsoft Docs
Extend visuals with report page tooltips - Power BI Microsoft Docs

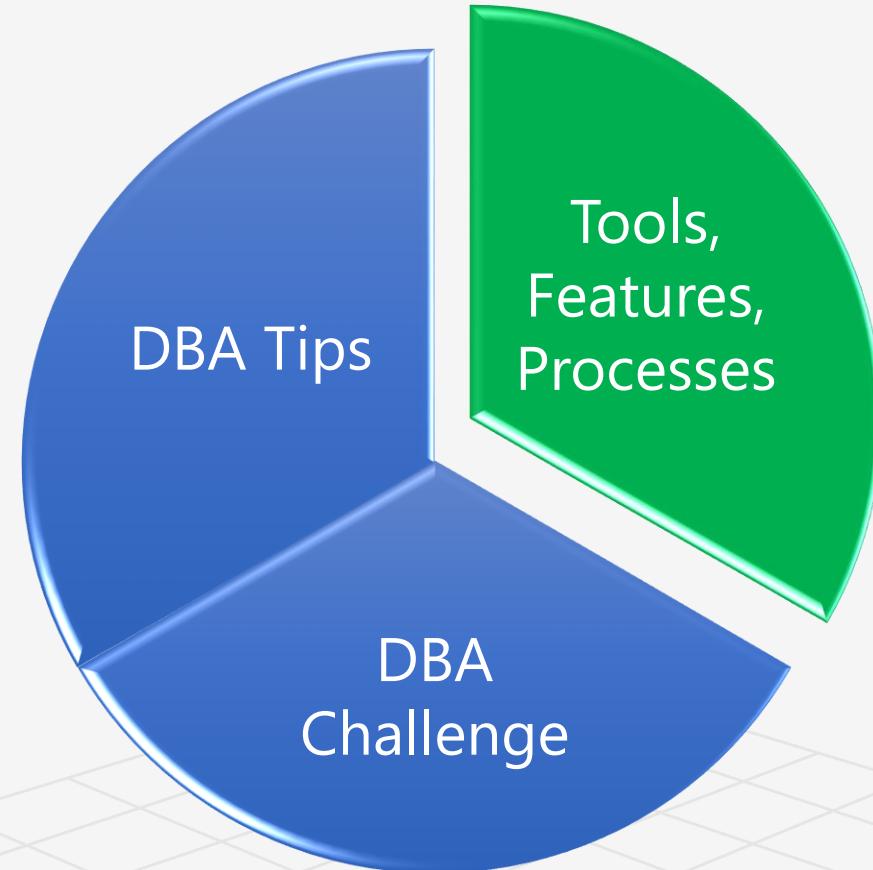
Data Development

9

Workflow Orchestration

- SQL Agent and Powershell/Python is a good combination
- Entire Azure Workday to On-Prem Active Directory Onboarding Solution in pure Powershell and SQL Server
 - Integrates Employees with 6 other downstream systems
 - Creates Exchange Mailboxes
 - Manages AD Security Group/Distribution List Memberships
 - Manages and avoids Duplicate AD Login Name creation
- Replaced a \$5,000/yr Vendor app

Database Management



Database Management

Tools, Features and Processes

- SQL Server is an “Instrumented” application
- Tools:
 - Windows Resource Monitor, PerfMon, SSMS
- SQL Features:
 - SQL Agent Alerts
 - Extended Event Sessions
 - Query Store
 - DBCC CheckDB
 - Automate SQL Upgrades
 - Encryption
- Processes:
 - Resolve Locking/Blocking/Deadlocks
 - Table Partitioning
 - SQL Utility Database Maintenance
 - Reading Query Plans



Database Management

1

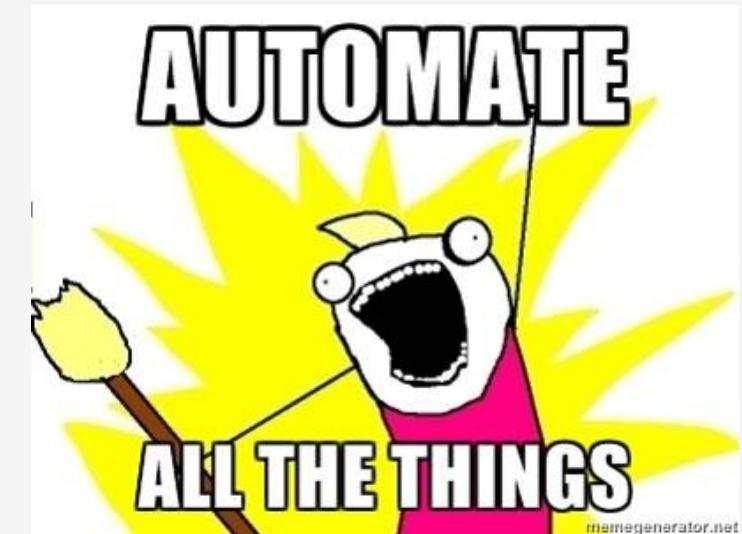
SQL Agent Alerts

- Alerts sent as emails DBMail
- Enable a standard set of Alerts
 - 833 – SAN too slow (15 Second Write)
 - 18056 – Bad Login
 - 17890 – SQL Out of Memory
 - 2812 – Stored Procedure Not found
 - 515 – Null INSERT attempted on not-null column
- Log them to a central Database for Trend Analysis

2

Automate Database Integrity Checks

- Yes, Yes we can
- Create a Separate SQL Server, restore backups and run DBCC CheckDB
- Capture the output results and Email to yourself



Database Management

3 Automate "Touchless" SQL Upgrades

- Use a Configuration.INI file with SETUP.EXE and a bit of Powershell

4 Extended Event Sessions

Replaced SQL Profiler/Server Trace in 2012

- Standard 5 to deploy:
 - Blocked Process Report
 - Deadlocks
 - Failed Logins
 - Good Logins
 - File Growth Ring Buffer
- <https://gist.github.com/gwalkey/5628793ed34ad1b5d54ed1a1a92f4780>

Database Management

5

Encryption

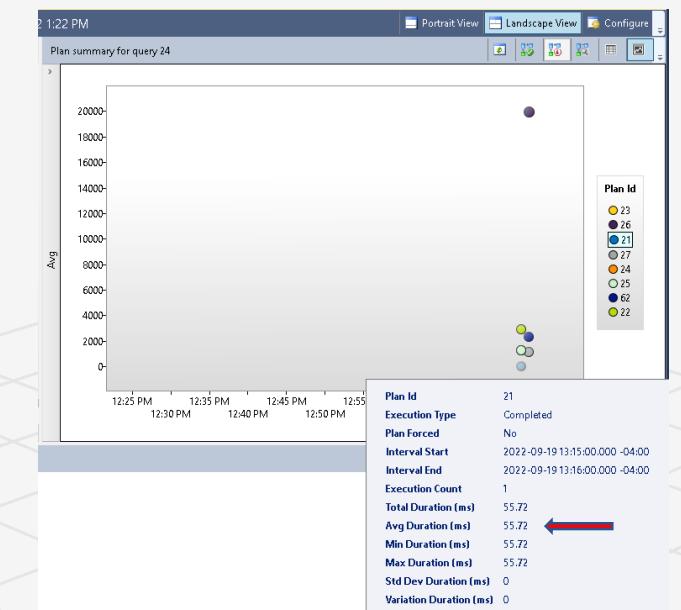
- TDE = Transparent Data Encryption
 - Data at rest (MDF/LDF/NDF/TempDB)
 - Encrypted Backups
- TDE and Wire/Protocol Encryption – both included Standard Edition since SQL 2019
- You can mint internal self-signed SSL Certificates for TDE - but not Best Practice
- If TDE SSL Cert expires: SQL keeps on rolling
- If Wire/Protocol SSL Cert expires: Can't Login!
- Must manage your PKI

Database Management

6

Query Store

- Debuted in SQL 2016
- Enable it everywhere* - Enabled by default in Azure
- Will capture Everything:
 - Queries to itself, Backup Sessions, Update Table Statistics
- Allows you to “Pin” or Force a good Query Plan ID
- Hack Query Store:
 - Build Multi-Server performance dashboards using QS
 - SQL Agent Job - “SQL Running for more than 30 seconds”



Database Management

6

Query Store Management

- DISABLE when Doing Heavy ETL, Re-Indexing, Schema Changes
 - ETLs create lots of Locks on Query Store's own Tables
- ALTER DATABASE [Database] SET QUERY_STORE = OFF - Pause
- ALTER DATABASE [Database] SET QUERY_STORE = ON - Enable
- ALTER DATABASE [Database] SET QUERY_STORE CLEAR - Wipe



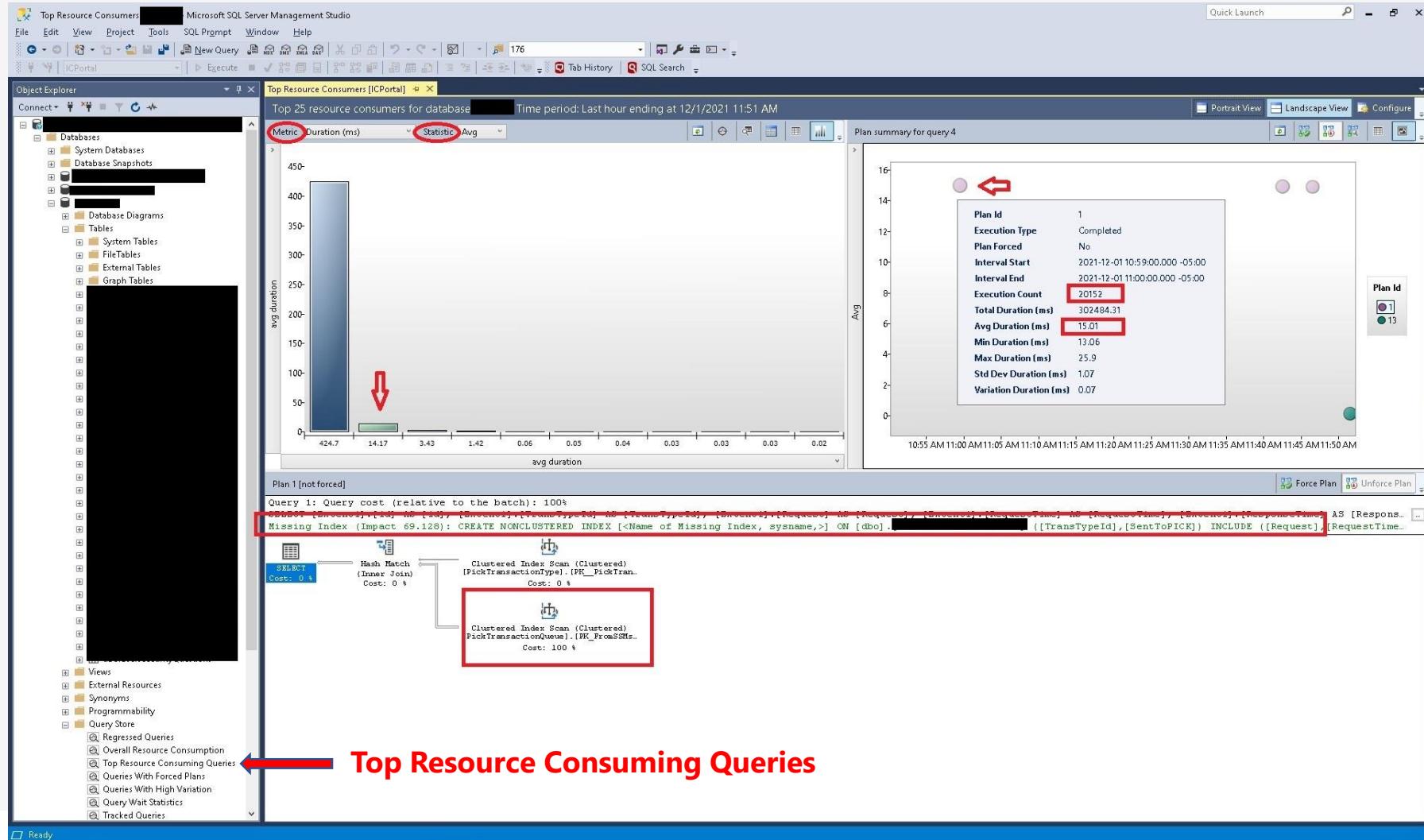
6

Top 100 Slowest Queries from Query Store

- <https://gist.github.com/gwalkey/3e9b4cc06dd7f0a48b2a01aafb70adef>

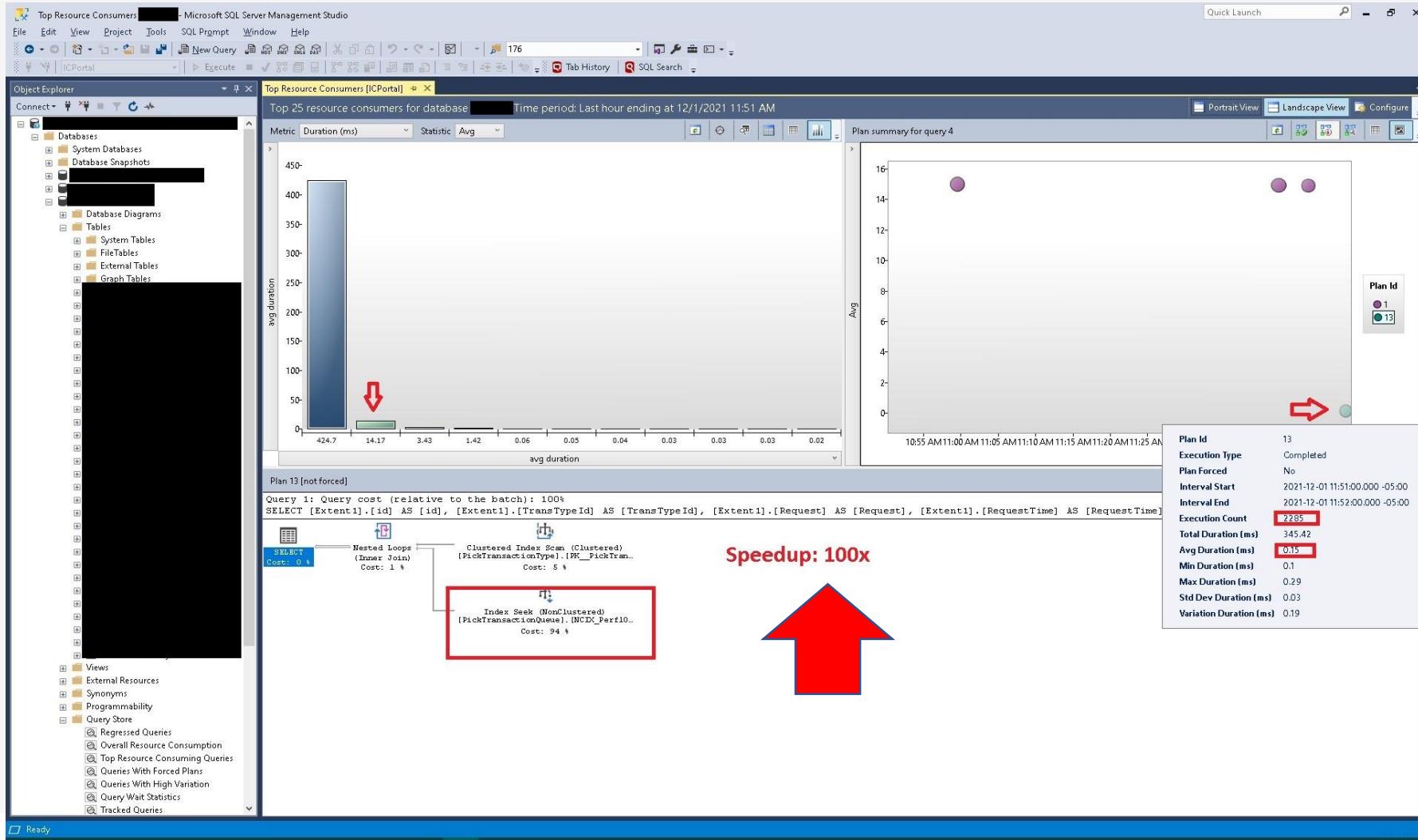
Database Management

Query Store Use Case - Find a Slow Query to Tune



Database Management

Query Store Use Case - Post-Tuning Results



Database Management

7

Concurrency

- Locking/Blocking/Deadlock Issues
- Pessimistic (Locks) vs Optimistic (Version Store/RCSI)
- Enable Blocked Process report
 - EXEC sp_configure 'blocked process threshold (s)', 5;
 - RECONFIGURE;
- Enable BPR Extended Event Session
- Enable Deadlock Extended Event Session
- RCSI is your friend - Optimistic Concurrency
- RCSI creates Row Versions in TempDB – no locks
- MS DTC is your enemy
 - DTC Transaction Isolation Level of Serializable will kill
- Concurrency Performance Chasing: Get off the row AFAP



Database Management

8

Table Partitioning

- Partitioning doesn't buy you better Query performance
 - except when locking get involved, then it does
- Lock escalations to table level can cause issues on large tables
- Generally used to rotate data in/out or for large tables
- Select Partition Column - typically a Business Logic Bucket (Year)
- Set Lock Escalation Level to Auto (Partition)
- TRUNCATE TABLE Product WITH (PARTITIONS (1))
- Truncates Aligned Indexes too

Database Management

9

System Database Cleanups

These MSDB tracking tables can GROW LARGE and should be maintained

- Database Mail History
- Database Backup History
- SQL Agent Job Execution History
- SSIS Package Execution History

Use System Stored Procedures to clean them out

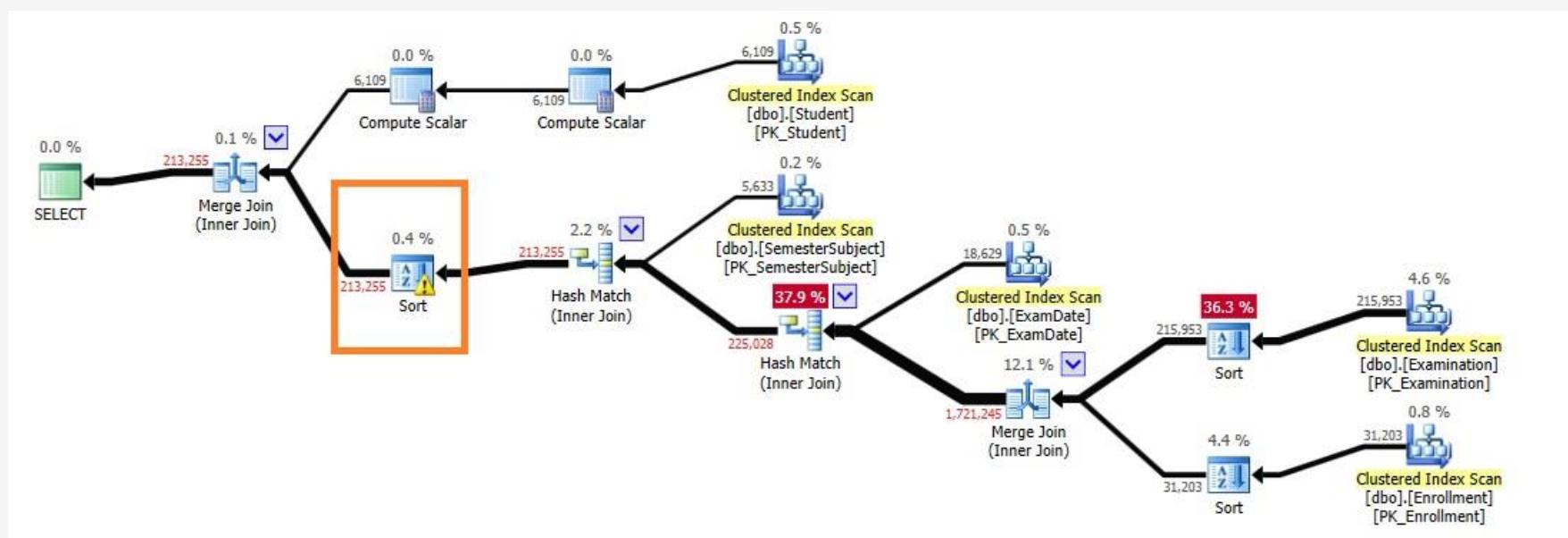
Database Management

A

Reading Query Plans

A Query Plan is a map of the SQL Engine's access path taken to get your answer

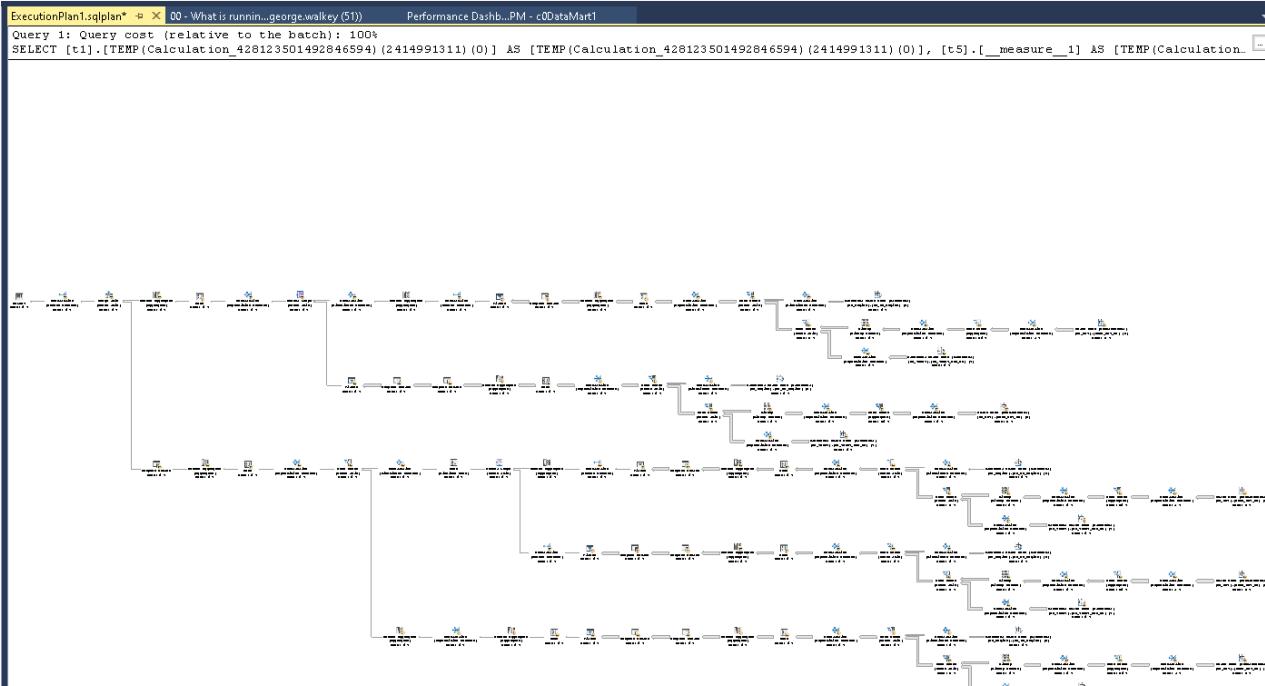
- Includes detailed info on row counts, data movement, durations
- Shows Indexes, Operators, Splits/Joins, Parallelism, TempDB spills
- Starts on the right, flows to the left
- Your Job as a DBA is to ID a “Bad Plan” and make it a Good or Better Plan



Database Management

A

Reading Query Plans – Bad?



DataDev Tips

1 Monitoring Scripts that will help you out

- Show What's Running Now with Query Plans

<https://gist.github.com/gwalkey/3969e5b4389b67987eead7e9e36fb9f4>

- Show Current Locks (Database/Table/Extent/Page/Row/Key/Partition)

<https://gist.github.com/gwalkey/8b53eeec52b1eb627ac7835ac89293c4>

- Index CRUD Performance Breakdown (Scans vs Seek)

<https://gist.github.com/gwalkey/9a6da12a9b0a8b7b4815ee3db9ef2b65>

- Connections Per Database

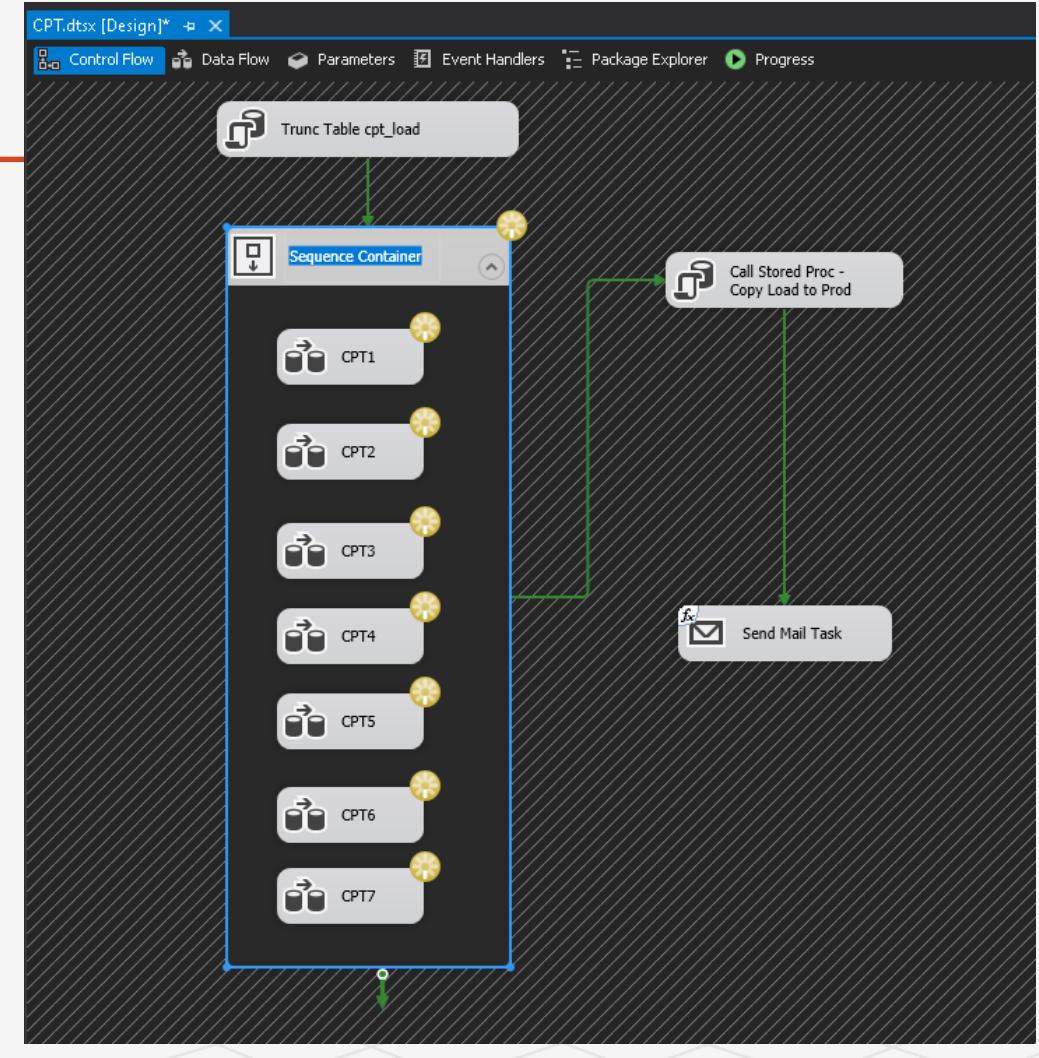
<https://gist.github.com/gwalkey/763d214febb8bf2601b6d32380d02beb>

- SET Statistics TIME, IO, (PROFILE) ON

- Shows Total Logical/Physical Reads, Read-Aheads, Time Duration, (Exec Tree)

DataDev Tips

2 SSIS – Load Multiple Files at the same time



Show Whats Running Now with Query Plans:

	Spid	BlockedBy	Database	HostName	User	Command	TrnIsolLevel	Dop	OpenTrans	CPUTime	Logical Reads	reads	writes	Rows	MemGrantKB	Status	WaitType	Application	QueryPlan	SqL Stmt
1	60	0			PF\george.walkey	BULK INSERT	ReadCommitted	1	2	1279	11090	4	7274	1	0	suspended	ASYNC_NETWORK_ID	Microsoft SQL Server	<ShowPlanXML xmlns="http://schemas.microsoft.com/	<?query - in
2	62	0			PF\george.walkey	BULK INSERT	ReadCommitted	1	2	1334	9530	0	6233	1	0	suspended	ASYNC_NETWORK_ID	Microsoft SQL Server	<ShowPlanXML xmlns="http://schemas.microsoft.com/	<?query - in
3	68	0			PF\george.walkey	BULK INSERT	ReadCommitted	1	2	1081	9745	0	6420	1	0	suspended	ASYNC_NETWORK_ID	Microsoft SQL Server	<ShowPlanXML xmlns="http://schemas.microsoft.com/	<?query - in
4	71	0			PF\george.walkey	BULK INSERT	ReadCommitted	1	2	1667	11274	0	7437	1	0	suspended	ASYNC_NETWORK_ID	Microsoft SQL Server	<ShowPlanXML xmlns="http://schemas.microsoft.com/	<?query - in
5	73	0			PF\george.walkey	BULK INSERT	ReadCommitted	1	2	1016	8021	0	5229	1	0	running	NULL	Microsoft SQL Server	<ShowPlanXML xmlns="http://schemas.microsoft.com/	<?query - in
6	74	0			PF\george.walkey	BULK INSERT	ReadCommitted	1	2	1024	9526	0	6226	1	0	suspended	ASYNC_NETWORK_ID	Microsoft SQL Server	<ShowPlanXML xmlns="http://schemas.microsoft.com/	<?query - in
7	75	0			PF\george.walkey	BULK INSERT	ReadCommitted	1	2	1378	9529	0	6227	1	0	suspended	ASYNC_NETWORK_ID	Microsoft SQL Server	<ShowPlanXML xmlns="http://schemas.microsoft.com/	<?query - in

DataDev Tips

3

JSON Parsing

- Import JSON Documents into SQL LOAD Table with varchar(max) column
- INSERT INTO Prod SELECT FROM Load with OUTERAPPLY OPENJSON()

```
{  
  "MedicalRecords": [  
    {  
      "MedicalRecord": [  
        "MedicalRecordID": "SomeIDNumber",  
        "VisitDate": "08.32.2020",  
        "BroughtInBy": "Mother",  
        "Condition": "Alert/Amb",  
        "Remarks": "Perm to TX- Yes/Trusted User/Mother/sps.",  
        "HistoryBy": "Mother",  
        "Complaint": "UTI",  
        "Occupation": "Student",  
        "Tetanus": "<5",  
        "TakingMedications": [  
          {  
            "TakingMedication": "PREDNISONE TAB 10M(00591544210)",  
            "Initials": "gbw27"  
          },  
          {  
            "TakingMedication": "TRIAMCINOLONE 0.1% CREAM 30GM(51672120202)",  
            "Initials": "gbw27"  
          },  
          {  
            "TakingMedication": "CLORETASOL CRE 0.0(68180095604)",  
            "Initials": "gbw27"  
          }  
        ],  
        "BloodPressureMeasurements": [  
          {  
            "Value": "100/70",  
            "MethodTaken": "ReM",  
            "TimeTaken": "22:31:17",  
            "Initials": "gbw27"  
          }  
        ],  
        "PulseRateMeasurements": [  
          {  
            "Value": "89",  
            "Irregular": "",  
            "TimeTaken": "22:31:17",  
            "Initials": "gbw27"  
          }  
        ],  
        "RespirationMeasurements": [  
          {  
            "Value": "16",  
            "TimeTaken": "22:31:17",  
            "Initials": "gbw27"  
          }  
        ]  
      ]  
    ]  
  ]  
}
```

ID	MRID	DocumentID	CenterID	VisitDate	BroughtInBy	Condition	Remarks	HistoryBy	Complaint	Occupation	Tetanus	Pregnant	Nursing	LastMenstrualPeriodRecorded
1	97893	NULL	1	2021-08-21	Self	Alert/Amb	NULL	Self	COVID-19 Test(Asx)	Retired	<5	NULL	NULL	NULL
2	97894	NULL	1	2021-08-21	Self	Alert/Amb	NULL	Self	COVID-19 Test(Asx)	Marketing	<5	N	N	08.01.21
3	97895	NULL	1	2021-08-21	Self	Alert/Amb	NULL	Self	Runny Nose/Covid Test	Election Office	?	NULL	NULL	NULL
4	97896	NULL	1	2021-08-21	Self	Alert/Amb	NULL	Self	Persistent Coughing	Sales	?	NULL	NULL	NULL
5	97897	NULL	1	2021-08-21	Daughter	Alert/Amb	PF is not PCP	Self	Chest Discomfort/Dizziness	Supervisor	<5	NULL	NULL	NULL
6	97898	NULL	1	2021-08-21	Self	Alert/Amb	NULL	Self	?Sinus Infection	Teacher	<5	NULL	NULL	NULL
7	97899	NULL	1	2021-08-21	Father	Alert/Amb	Perm to TX-Yes/[REDACTED]/Father/krs.	Father	Immunization	Student	?	N	N	08.20.21
8	97900	NULL	1	2021-08-21	Self	Alert/Amb	NULL	Self	COVID-19 Test(Asx)	Executive Recruiting	<5	N	N	2021
9	97901	NULL	1	2021-08-21	Self	Alert/Amb	NULL	Self	Fall/R Knee Pain/Back Pain	Retired/NA	?	NULL	NULL	NULL
10	97902	NULL	1	2021-08-21	Self	Alert/Amb	Employer Employment Verified by UTR/[REDACTED]	NULL	FT-Drug Screen	NULL	NULL	NULL	NULL	NULL
11	97903	NULL	1	2021-08-21	Self	Alert/Amb	NULL	Self	COVID-19 Test(Asx)	Shift Supervisor	?	NULL	NULL	NULL
12	97904	NULL	1	2021-08-21	Self	Alert/Amb	NULL	Self	H/Runny Nose/Covid Test	Maintenance	?	NULL	NULL	NULL
13	97905	NULL	1	2021-08-21	Self	Alert/Amb	NULL	Self	?R Hand Finger Injury	RN	?	NULL	NULL	2016
14	97906	NULL	1	2021-08-21	Self	Alert/Amb	NULL	Self	Bodyache/Fatigue	Banker	>5	NULL	NULL	NULL
15	97907	NULL	1	2021-08-21	Mother	Alert/Amb	PF is not PCP Perm to TX-Yes/[REDACTED] Mother/	Mother	Neck Pain	Student	?	NULL	NULL	NULL
16	97908	NULL	1	2021-08-21	Mother	Alert/Amb	Perm to TX-Yes/[REDACTED] Mother/mf.	Mother	Immunizations	Student	>5	N	N	08.16.21
17	97909	NULL	1	2021-08-21	Self	Alert/Amb	NULL	Self	COVID Testing?Head ache/Body Ache/Shoriness/Fever	Homemaker	<5	N	N	08.20.21
18	97910	NULL	1	2021-08-21	Father	Alert/Amb	Employer Perm to TX-Yes/[REDACTED]/Father Empl	Father	L Hand Burn	Student/Cook	?	NULL	NULL	NULL
19	97911	NULL	1	2021-08-21	Self	Alert/Amb	NULL	Self	Personal	Forklift Operator	<5	N	N	08.21.21
20	97912	NULL	1	2021-08-21	Self	Alert/Amb	Employer	NULL	FT-Drug Screen	NULL	NULL	NULL	NULL	NULL
21	97913	NULL	1	2021-08-21	Self	Alert/Amb	NULL	Self	(MM) Personal	Acupuncturist	<5	NULL	NULL	NULL
22	97914	NULL	1	2021-08-21	Self	Alert/Amb	NULL	Self	?Covid Infection	Disability	?	NULL	NULL	NULL

DataDev Tips

3

JSON Parsing - Schema On Read

```
INSERT INTO.....  
SELECT  
    M.id,  
    DisabilityInfo.[Description],  
    DisabilityInfo.BeginDate,  
    DisabilityInfo.LightDuty,  
    DisabilityInfo.FullDuty,  
    DisabilityInfo.MedicationAffectWork,  
    DisabilityInfo.Comment,  
    DisabilityInfo.Restriction,  
    DisabilityInfo.Capabilities  
FROM  
    load.medicalrecord M  
    OUTER APPLY OPENJSON(M.DisabilityInformation)  a varchar(max) column that has JSON in it  
    WITH (  
        [Description] VARCHAR(255)      '$.Description',  
        BeginDate VARCHAR(15)         '$.BeginDate',  
        LightDuty VARCHAR(255)        '$.LightDuty',  
        FullDuty VARCHAR(255)         '$.FullDuty',  
        MedicationAffectWork VARCHAR(150) '$.MedicationAffectWork',  
        Comment VARCHAR(255)          '$.Comment',  
        Restriction NVARCHAR(MAX) '$.Restriction' AS JSON,  
        Capabilities NVARCHAR(MAX) '$.Capabilities' AS JSON  
    ) AS DisabilityInfo
```



DataDev Tips

4

.NET SqlDataReader in PowerShell – Alternative to Linked Servers

```
# Set Source as Azure SQL DB
$SQLSrcConnectionString = "Data Source=server.database.windows.net;Initial Catalog=Products;UID=SomeUser;Password=SomePassword;Application Name=PowerShell Data Copier;encrypt=yes;Connection Timeout=60"
$SQLSrcCmd      = "SELECT ID,WhenRowAdded FROM dbo.Products order by ID"
$objSqlSrcConn = New-Object System.Data.SqlClient.SqlConnection $SQLSrcConnectionString
$objSqlSrcCmd  = New-Object System.Data.SqlClient.SqlCommand($SQLSrcCmd, $objSqlSrcConn)
try
{
    $objSqlSrcConn.Open()
}
catch
{
    Throw('Error Connecting to Azure SQL DB - Error is:[{0}]' -f $error[0])
}

# Do Magic Part
[System.Data.SqlClient.SqlDataReader]$SqlReader = $objSqlSrcCmd.ExecuteReader()

# Destination is Bulk Copy
[string]$CnnStrTarget = "Data Source=InternalServer;Integrated Security=SSPI;Initial Catalog=Products;Application Name=PowerShell Bulk Inserter"
try
{
    $SqlBulkCopy      = New-Object -TypeName System.Data.SqlClient.SqlBulkCopy($CnnStrTarget)
    $SqlBulkCopy.EnableStreaming      = $true
    $SqlBulkCopy.DestinationTableName = 'Products_Load'
    $SqlBulkCopy.BatchSize           = 10000 # rows
    $SqlBulkCopy.BulkCopyTimeout     = 0 # seconds, 0 (zero) = no timeout limit
    $SqlBulkCopy.WriteToServer($SqlReader)
}
catch [System.Exception]
{
    $_.Exception | Write-Output
}

# Cleanup SQL Objects
$SqlReader.Close()
$objSqlSrcConn.Close()
$objSqlSrcConn.Dispose()
$SqlBulkCopy.Close()
```

DataDev Tips

5

Visual Studio Database Projects

- Allows you to put your Database under Source Control and “Check it in”
- Allows you to do a Diff between versions and Rollback*
- If all Developers use it: no Schema Surprises at Production
 - AKA no sneaky out-of-band changes via SSMS
- Can Export the DB Project as a DACPAC file and deploy on-prem
- Schema Diff (Server to Server or Server to Model)
- Data Compare (Server to Server)
- Integrates with Azure DevOps Pipelines for CI/CD
- Must be able to “Build” the DB Project in order to Deploy it
- Visual Studio Complier finds many Database Errors during Build phase
 - Found Calls to non-existent Stored Procedures

DBA Challenge #1

1

How would you find and Fix this issue?

- SQL Query brought 4-Core 64GB Server to its knees
- Multiple Users started running the same Query
- DBA killed all the SQL processes
- DBA waited 10 minutes for them all to stop
- .NET Function doing **string split** on a column (9us)
- SQL 2014 – No Query Store
- Circa 2014 - SAN with spinning hard drives



[Joe Chang's ExecStats Tool](#)

DBA Challenge #1

- 1 Issue: Query Engine was running the CLR Scalar function AND Scanning

SET STATISTICS PROFILE ON

	BatchSeq	EstRows_redux	Stmt Text
1	35172	1.74573409557343	Sort(ORDER BY:([PatientAccounts].[dbo].[RemitItem].[LastName] ASC, [PatientAccounts].[dbo].[RemitItem].[FirstName] ASC, [PatientAccounts].[dbo].[RemitItem].[CenterNumber] ASC, [PatientAccounts].[dbo].[RemitItem].[RemitItemId] ASC))
2	35172	1.74573409557343	—Compute Scalar(DEFINE:([Expr1014]=CASE WHEN [Expr1011] IS NULL THEN NULL ELSE (1) END))
3	35172	1.74573409557343	—Parallelism(Gather Streams)
4	35172	1.74573409557343	—Nested Loops(Left Outer Join, OUTER REFERENCES:([PatientAccounts].[dbo].[RemitItem].[RemitItemId]))
5	35172	1	—Parallelism(Distribute Streams, RoundRobin Partitioning)
6	35172	1	— —Top(TOP EXPRESSION:((10)))
7	35172	1	— —Filter(WHERE:([Expr1004]>(0)))
8	35172	1	— —Sequence Project(DEFINE:([Expr1004]=row_number))
9	35172	1	— —Segment
10	35172	1	— —Compute Scalar(DEFINE:([Expr1003]=(1)))
11	35172	1	— —Sort(ORDER BY:([PatientAccounts].[dbo].[RemitItem].[LastName] ASC, [PatientAccounts].[dbo].[RemitItem].[FirstName] ASC, [PatientAccounts].[dbo].[RemitItem].[CenterNumber] ASC, [PatientAccounts].[dbo].[RemitItem].[RemitItemId] ASC))
12	35172	1	— —Compute Scalar(DEFINE:([PatientAccounts].[dbo].[RemitItem].[CenterNumber]=[PatientAccounts].[dbo].[RemitItem].[CenterNumber])))
13	35172	1	— —Parallelism(Gather Streams)
14	35172	1	— —Nested Loops(Inner Join, OUTER REFERENCES:([PatientAccounts].[dbo].[RemitItem].[RemitItemId]))
15	35172	1	— —Parallelism(Repartition Streams, RoundRobin Partitioning)
16	35172	1	— —Filter(WHERE:([PatientAccounts].[dbo].[RemitItem].[CenterNumber]=CONVERT_IMPLICIT(nvarchar, [PatientAccounts].[dbo].[GetCenterFromMRNO], 1)))
17	35172	31422420	— —Compute Scalar(DEFINE:([PatientAccounts].[dbo].[RemitItem].[CenterNumber]=[PatientAccounts].[dbo].[GetCenterFromMRNO](CONVERT_IMPLICIT(nvarchar(50), [PatientAccounts].[dbo].[GetCenterFromMRNO], 1))))
18	35172	31422420	— —Index Scan(OBJECT:([PatientAccounts].[dbo].[RemitItem].[IDX_RemitItem_MRNO]))
19	35172	1	— —Clustered Index Seek(OBJECT:([PatientAccounts].[dbo].[RemitItem].[PK_RemitItem_2])), SEEK:([PatientAccounts].[dbo].[RemitItem].[RemitItemId]=[PatientAccounts].[dbo].[RemitItem].[RemitItemId]))
20	35172	1.74573409557343	—Compute Scalar(DEFINE:([Expr1012]=[PatientAccounts].[dbo].[RemarkCode].[RemarkCodeId], [Expr1013]=[PatientAccounts].[dbo].[RemarkCode].[Description])))
21	35172	1.74573409557343	—Nested Loops(Inner Join, OUTER REFERENCES:([PatientAccounts].[dbo].[RemitItem].[RemarkCodeId]))
22	35172	1.74573409557343	—Compute Scalar(DEFINE:([Expr1011]=[PatientAccounts].[dbo].[RemitItem].[RemarkCodeId]))
23	35172	1.74573409557343	— —Index Seek(OBJECT:([PatientAccounts].[dbo].[RemitItem].[PK_RemitItem_2])), SEEK:([PatientAccounts].[dbo].[RemitItem].[RemarkCodeId]=[PatientAccounts].[dbo].[RemitItem].[RemarkCodeId]))
24	35172	1	— —Clustered Index Seek(OBJECT:([PatientAccounts].[dbo].[RemarkCode].[PK_RemarkCode])), SEEK:([PatientAccounts].[dbo].[RemarkCode].[RemarkCodeId]=[PatientAccounts].[dbo].[RemarkCode].[RemarkCodeId]))



SQL Server Concurrency

Locking, Blocking and Row Versioning

By Kalen Delaney



DBA Challenge #2

2

11M row table, 3 Minute Runtime

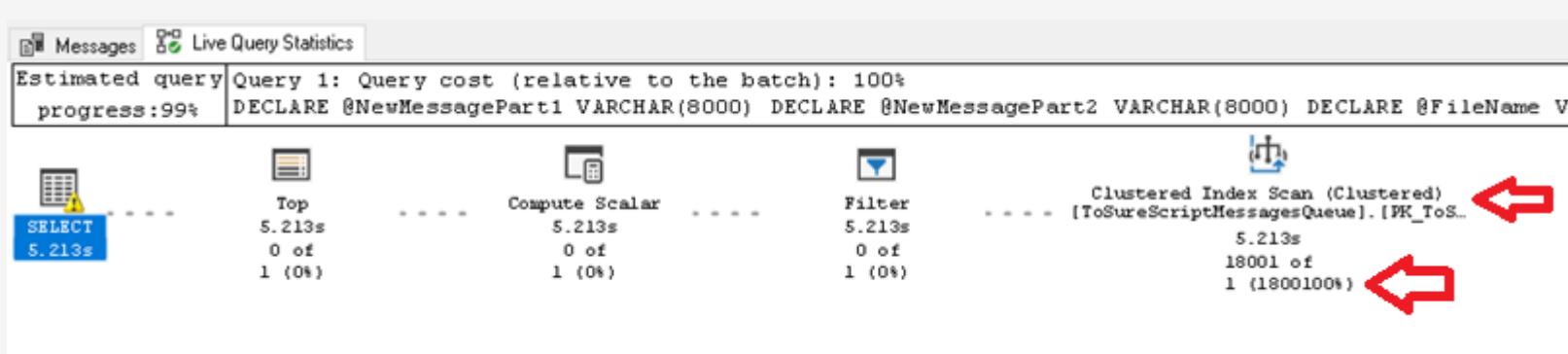
- Blocked everyone else during its execution

```
DECLARE      @NewMessagePart1 VARCHAR(8000)
DECLARE      @NewMessagePart2 VARCHAR(8000)
DECLARE      @FileName VARCHAR(50)

SELECT      TOP 1
            @NewMessagePart1 = CAST(Request AS VARCHAR(8000)),
            @NewMessagePart2 = SUBSTRING(Request, 8001, 8000),
            @FileName = filename
FROM        tosurescriptmessagesqueue
WHERE       request -- < Varchar(max)
LIKE        '%<MessageID>' + '14203a93c8b69e749ad14b6dd6aaf71c' + '</MessageID>%'
AND        transtypeid = 34
ORDER BY    MessageQueueID DESC
```

SET STATISTICS IO, TIME ON

- 10,351,648 logical reads,**
- 91 physical reads ,
- 9,943,094 read-ahead reads



DBA Challenge #2

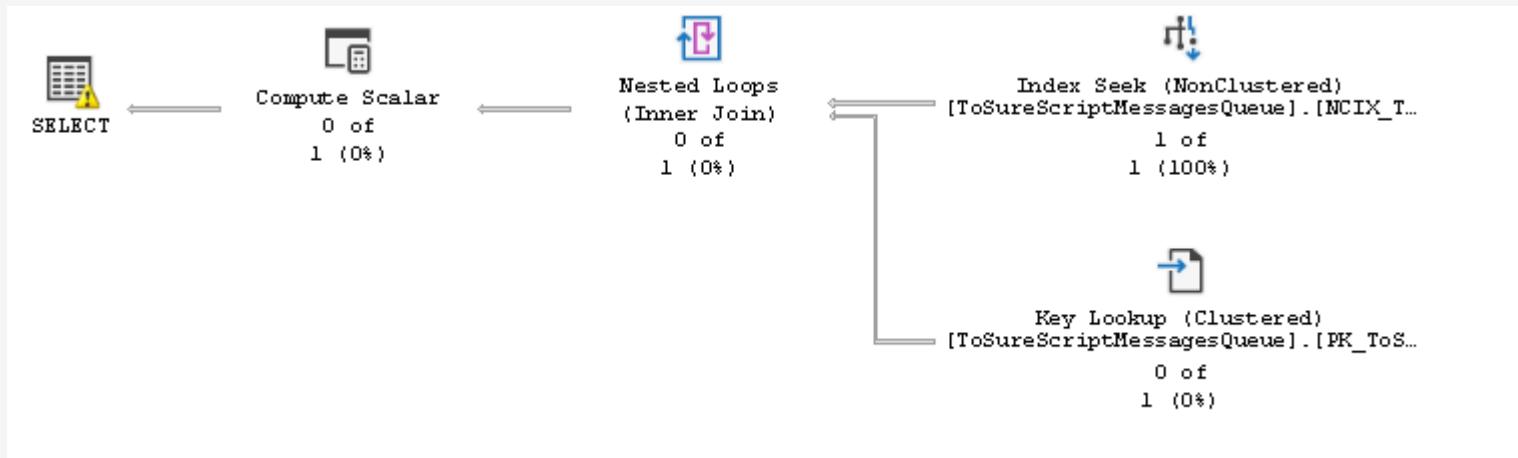
2

11M row table, **3ms** Runtime

- Didn't need RCSI – just Good Indexing

```
DECLARE  @NewMessage VARCHAR(MAX)
DECLARE  @OriginalMessage VARCHAR(MAX)
DECLARE  @FileName VARCHAR(50)

SELECT  @OriginalMessage = Request,
        @FileName = [FILENAME]
FROM    trowserscriptmessagesqueue
WHERE   MessageId='14203a93c8b69e749ad14b6dd6aaf71c'
        AND transtypeid = 34
```



SET STATISTICS IO, TIME ON

- **8 logical reads**
- 5 physical reads
- 0 read-ahead reads

570x
Performance
Improvement
And
No Blocking

DBA Challenge #3

3 Real-World Problem needed a Quick solution

SQL View using 5-way UNION and Linked Servers brought

- SQL View had a 5-way UNION with one local and 4 remote tables
- SQL View fronted a Developer Logging WebSite with multiple active users
- The View itself took over 3 minutes to pull all the data from the Remote servers
- Once the data volume in the Remote tables reached a certain row count
- The View Blocked 4 different apps on the remote SQL servers from being able to run
- Why?
- Because Linked Servers are DUMB!
 - They LOCK the ENTIRE remote table, pull it across the wire, THEN return it to the View
- "A few million rows here, a few million there, and now you're talking BILLIONS!"
- Solution #1 – Convert 5-way SQL UNION to use OPENQUERY with Remote Column Aliases
 - Trying to avoid locking on the remote servers - FAIL

SQL Handbooks



SQL Server Concurrency

Locking, Blocking and Row Versioning

By Kalen Delaney



The SQL Server Extended Development Environment



George Walkey

Patient First Corporation

<https://github.com/gwalkey>

George.walkey@patientfirst.com



The Next Slide is You