
# Conquering the Monster Proc: How to Combat Legacy Code

**Deborah Melkin**

She\Her

Data Engineer

Advisor360

# Deborah Melkin

She\Her

## Data Engineer

Advisor360

- 20+ years as a DBA
- NESQL Board Member
- Co-Founder, WITspiration
- Speaker Idol Winner 2019
- #Redgate100 (2022)
- Microsoft MVP. Data Platform


@dgmelkin

DebtheDBA.wordpress.com

dgmelkin@gmail.com


PASS Data Community SUMMIT 2023

# Solving Real World SQL Server Problems

*Learning Pathway*

**Wednesday**

- Unlocking the Power of WhoIsActive – *Jeff Iannucci*
- Is Storage the Root Cause of Your Performance Woes... or Not? – *Andy Yun*

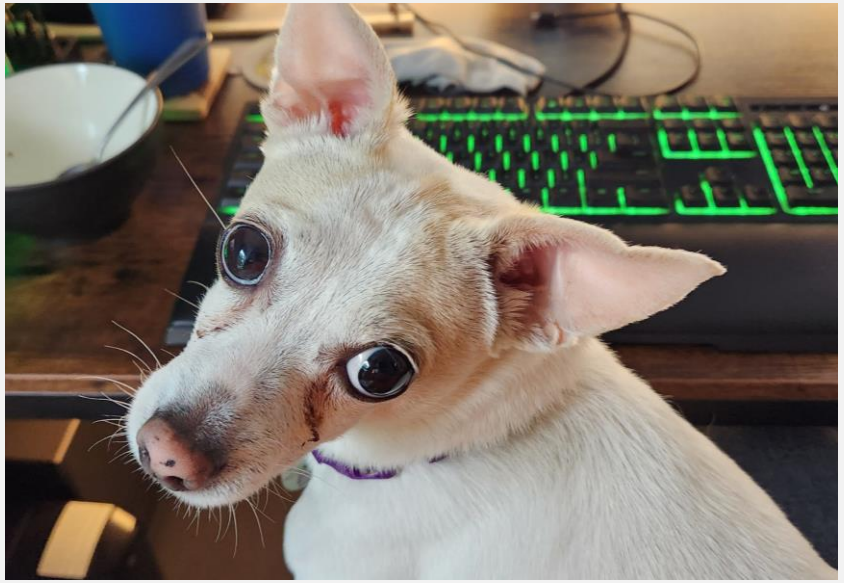
**Thursday**

- Mastering Dynamic SQL – *Deborah Melkin*
- Making the Most of Query Store in the Real World – *Jeff Iannucci*

**Friday**

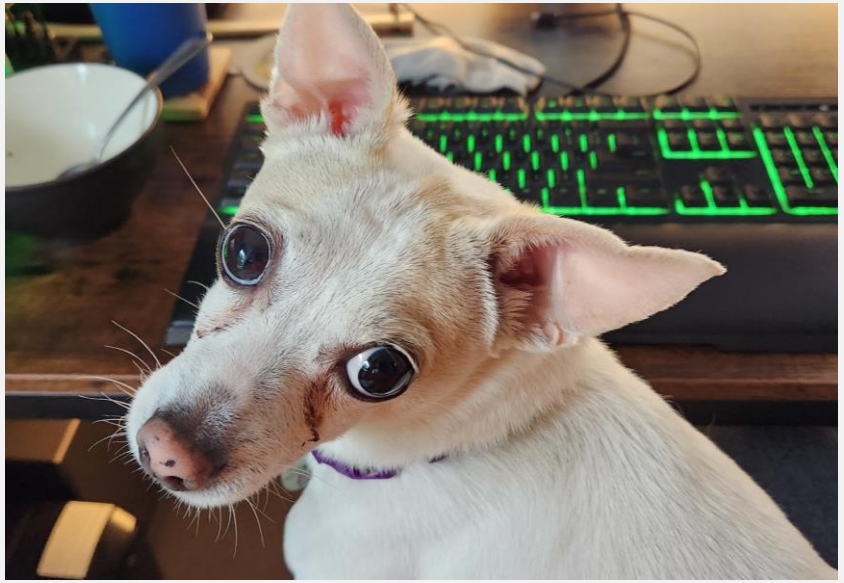
- Conquering the Monster Proc – How to Combat Legacy Code – *Deborah Melkin*

PASS Data Community SUMMIT 2023



# Story Time

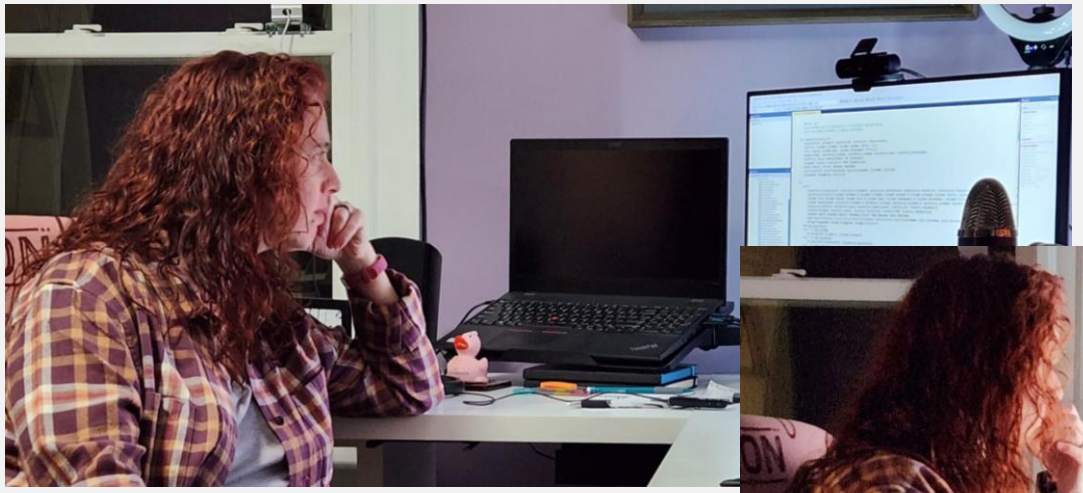
2 DBAs and a Developer walked into my office....

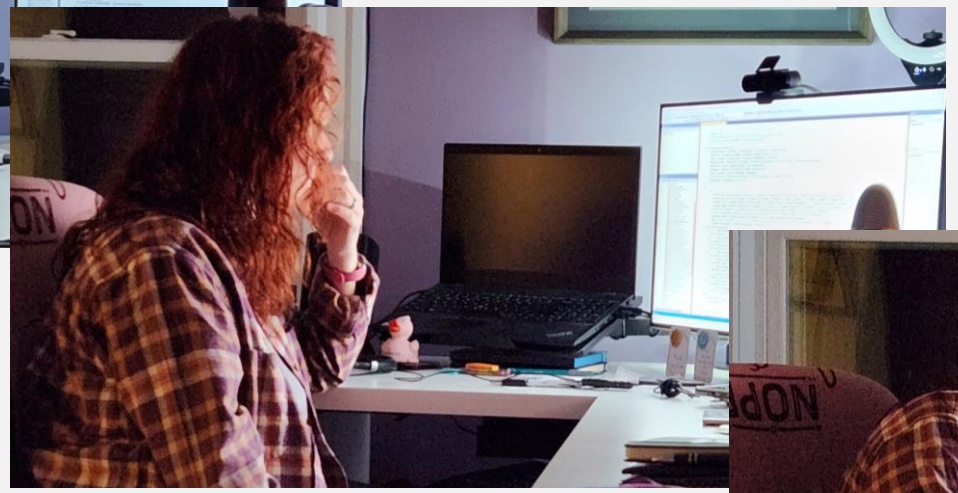


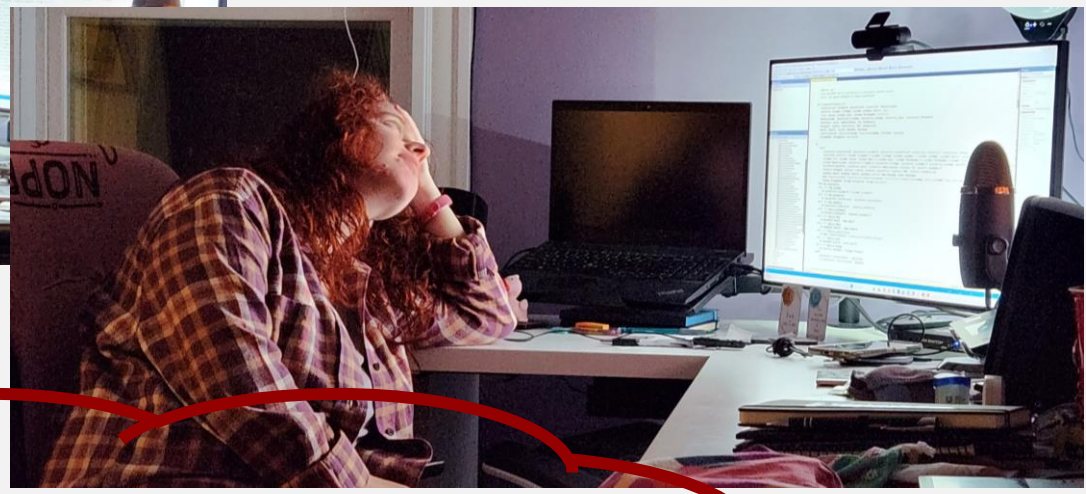
**Congratulations!**
**You know that proc everyone hates?**
**You finally get to fix it!**

PASS Data Community SUMMIT 2023

**Woo hoo!!!!**







**Oh wait, now what do I do?**

# Does this sound familiar?

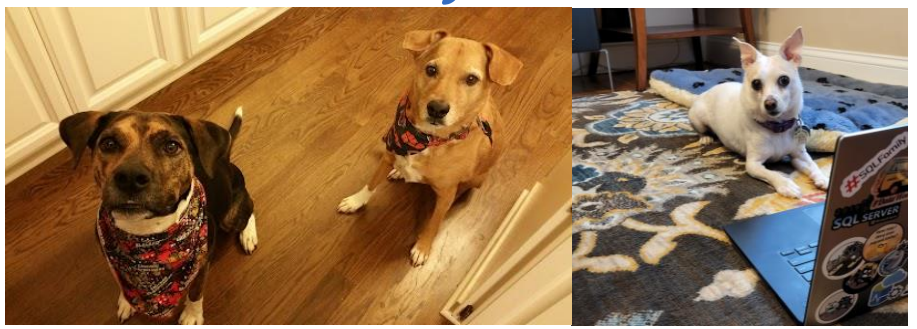
Our goal is to be smarter about how to attack these legacy code.

# Here’s the code we’re going through…

demo.sp_NightlyProcessingForReporting

# History of the Procedure

The Programmers



- Originally created in “2006”
- Modified by multiple programmers over the years
- Updates the demo.NightlySalesStaging table with the sales history for specified dates
- Creates the data needed for about 9 tables used by reports based on all of the data in staging table

# Here’s how we’re going to attack it…

# Two Plans of Attack to Make Changes

- Technical ← **Why**
- Business ← **Why Not**

# Technical Plan of Attack

# Technical Plan of Attack

- Gather requirements
- “Reverse Engineer” the code
- Gather performance information
- Determine the fix
- Design test plans

# Gather Requirements

- What problem are you trying to solve?
- What is the code supposed to do?
- Is the code currently doing what it's supposed to do?

# Gather Requirements – Q1 Answered

- What problem are you trying to solve?
  - *I've been told the performance is bad*
  - *It has some very involved logic that's hard to understand*

# Gather Requirements – Q2 Answered

- What is the code supposed to do?
  - *Load tables used by reports after a nightly staging process*
  - *Note hidden inside*

```
173     /*****************************************************************************
174     * Populate Reporting Tables
175     *
176     * 2006-10: Jess
177     * Each SSRS report will read from a vw_rpt_[name] view that will be a
178     * SELECT * from the respective rpt_[name] table
179     *
180     * IMPORTANT - All reports should only ever query demo.NightlySalesStaging
181     *****************************************************************************/
```

# Gather Requirements – Q3 Answered

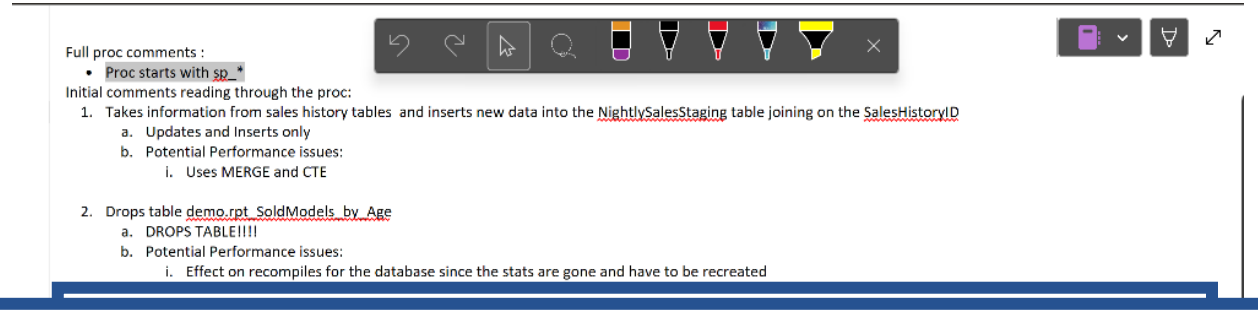
- Is the code currently doing what it's supposed to do?
  - *Code does update report tables*
  - *Not all query use only the demo·NightlySalesStaging table*

- *Any other technical questions that I've forgotten to ask?*

# Reverse Engineer the code

- What does each step do?
- Initial thoughts
- Review the code multiple times

# Reverse Engineer - Handwritten Notes



Full proc comments :

- Proc starts with sp_*

Initial comments reading through the proc:

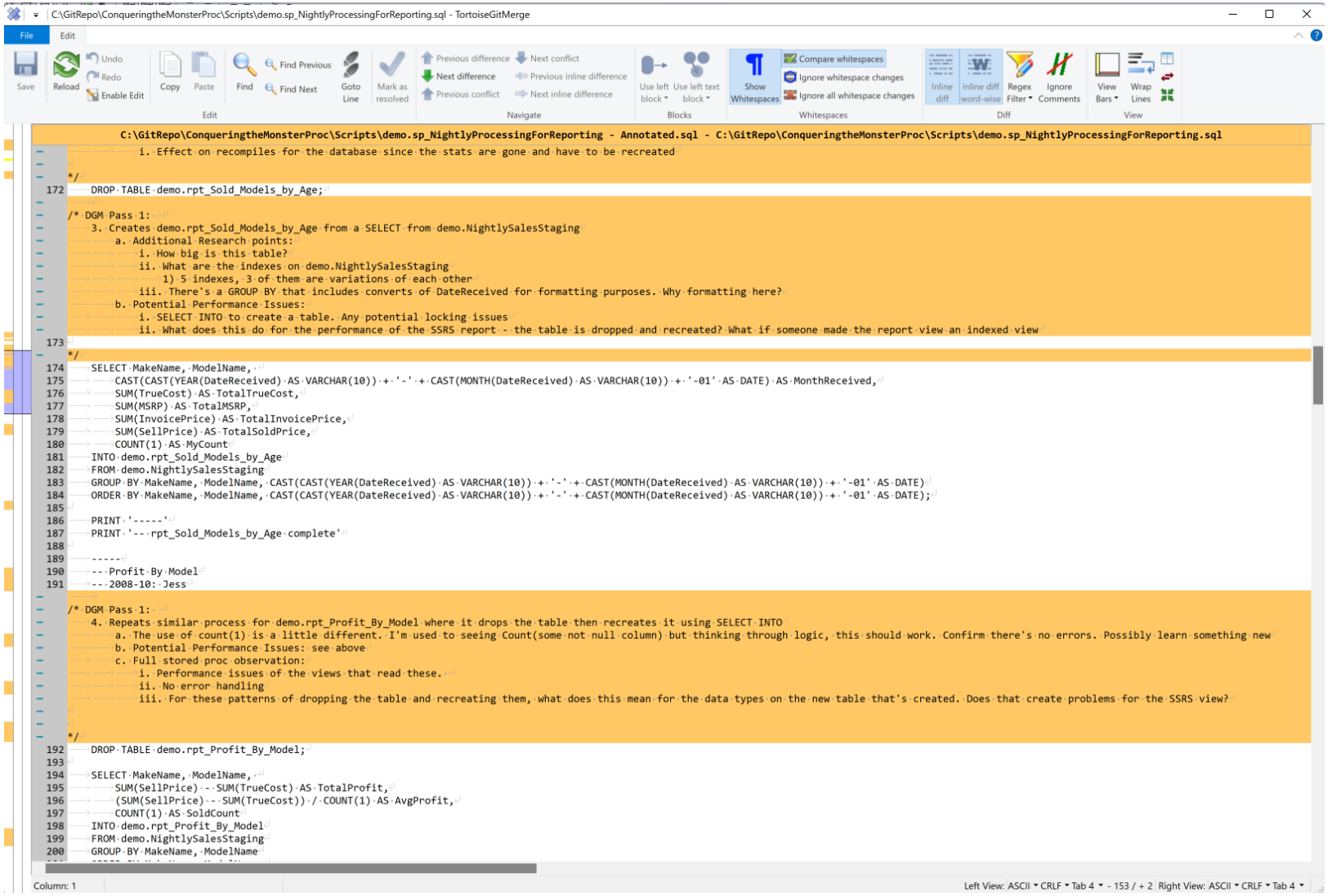
1. Takes information from sales history tables and inserts new data into the NightlySalesStaging table joining on the SalesHistoryID
   a. Updates and Inserts only
   b. Potential Performance issues:
      i. Uses MERGE and CTE

2. Drops table demo.rpt_SoldModels_by_Age
   a. DROPS TABLE!!!!
   b. Potential Performance issues:
      i. Effect on recompiles for the database since the stats are gone and have to be recreated

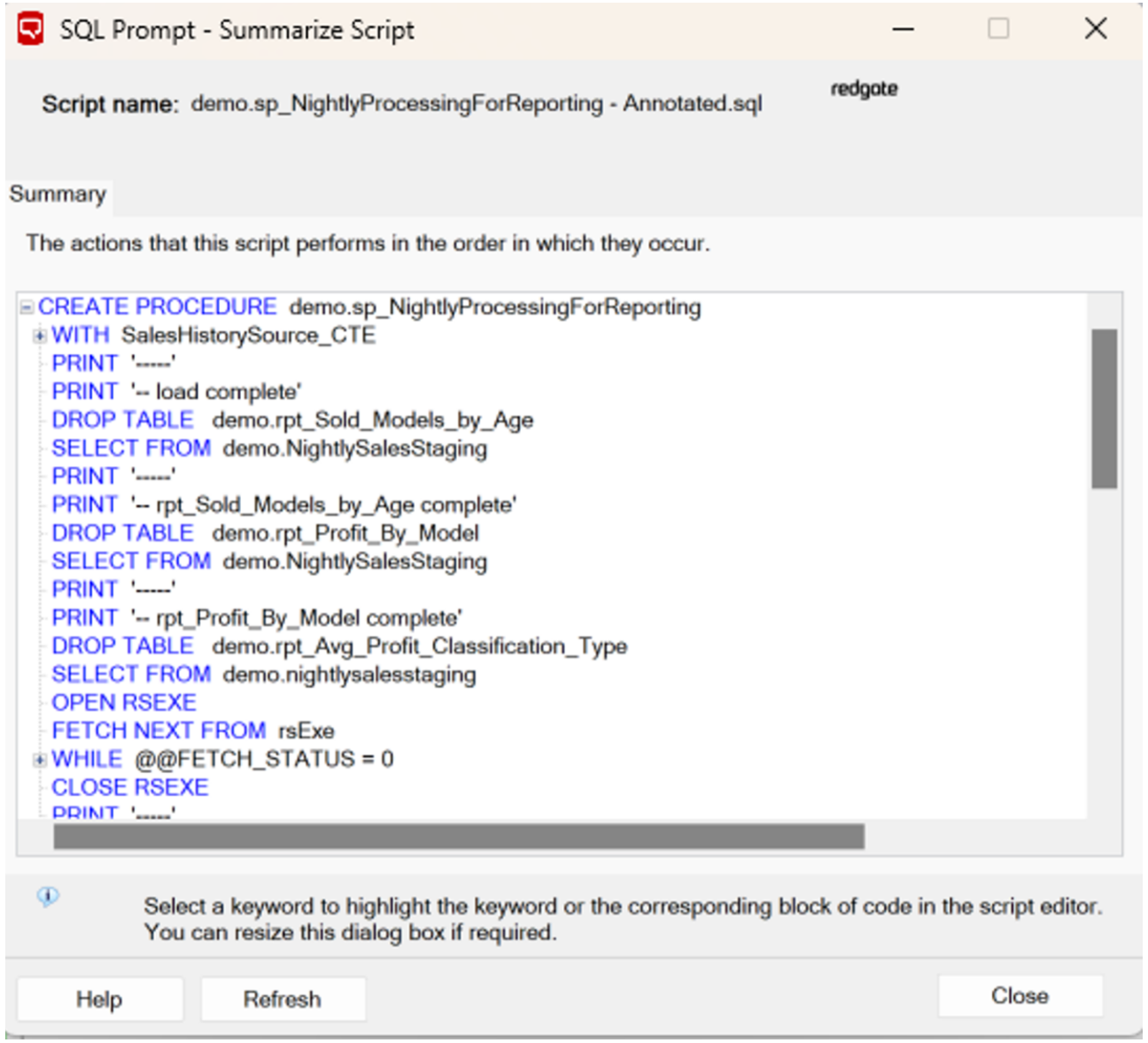
3. Creates demo.rpt_Sold_Models_by_Age from a SELECT from demo.NightlySalesStaging
   a. Additional Research points:
      i. How big is this table?
      ii. What are the indexes on demo.NightlySalesStaging
         1) 5 indexes, 3 of them are variations of each other
      iii. There's a GROUP BY that includes converts of DateReceived for formatting purposes.
   b. Potential Performance Issues:
      i. SELECT INTO to create a table. Any potential locking issues
      ii. What does this do for the performance of the SSRS report - the table is dropped and r

7. Same pattern for demo.rpt_salessummarypermonth
   a. NOTE - different developer (Sebastian), naming convention
   b. NOLOCKS!!!! :(
   c. Uses a temp table to hold data from sales person and sales history
      i. Still uses SELECT INTO *smh*
   d. Joins the temp table toa view to do a SELECT INTO the report table.
      i. Temp table doesn't have any indexes
      ii. View doesn't have a schema name so SQL Prompt is no help
         1) Confirmed it's in the dbo schema.

# ...Or Annotate Script



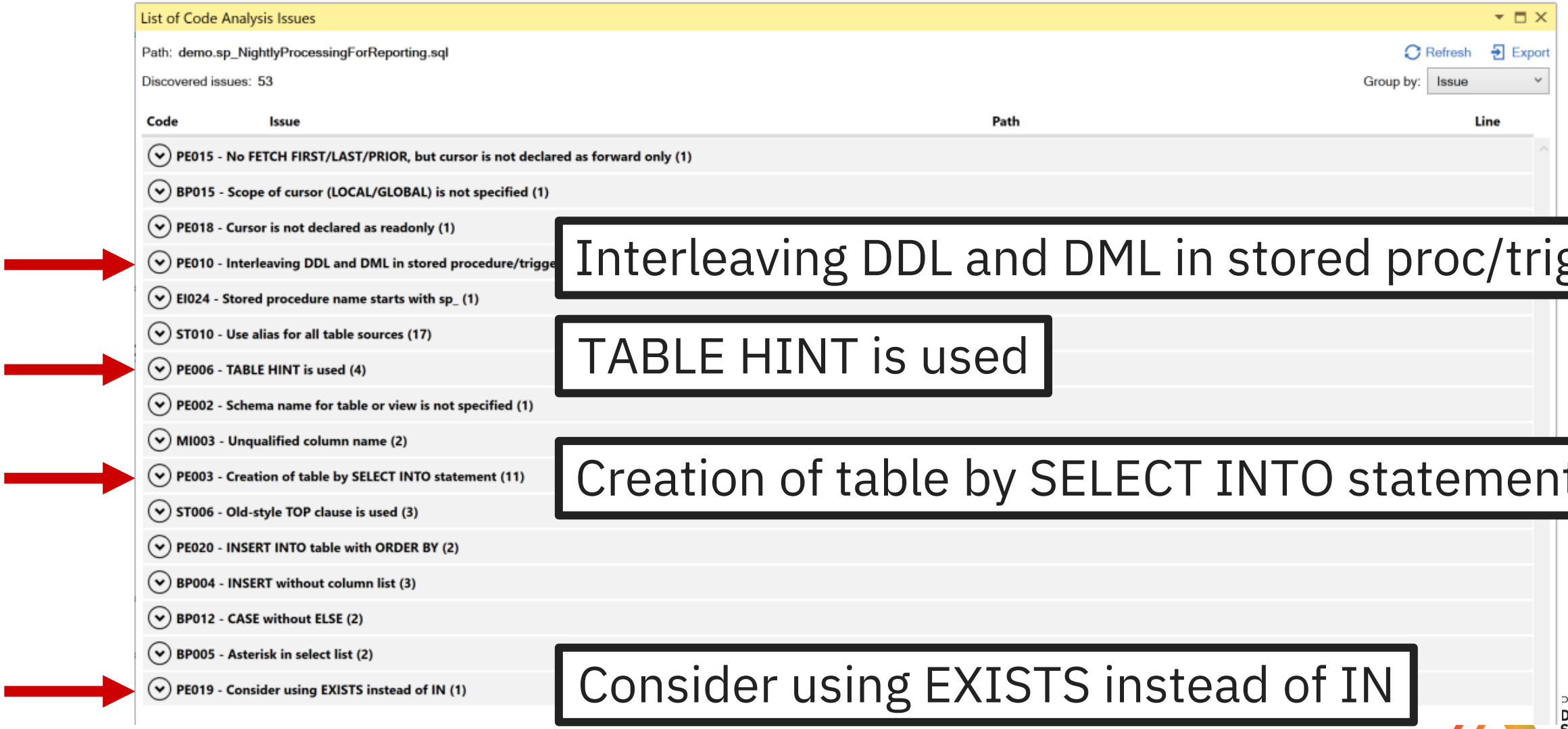
# SQL Prompt – Summarize Script



# Tools to help identify potential issues

- Code Analysis
  - ScriptDOM
  - SQLPrompt Code Analysis
  - Visual Studio – Analyze
- Look for Dependencies
  - sys.sql_dependencies
  - sp_helpExpandView (Not just for views!)

# SQL Prompt - Code Analysis




Interleaving DDL and DML in stored proc/trigger

TABLE HINT is used

Creation of table by SELECT INTO statement

Consider using EXISTS instead of IN

# Code Analysis – Visual Studio

```
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(465,3): Warning:  : SR0001 : Microsoft.Rules.Data : The shape of the result set produced by a SELECT * statement will change if the underlying table or view structu
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(510,8): Warning:  : SR0001 : Microsoft.Rules.Data : The shape of the result set produced by a SELECT * statement will change if the underlying table or view structu
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(72,4): Warning:  : SR0007 : Microsoft.Rules.Data : Nullable columns can cause final results to be evaluated as NULL for the predicate.
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(73,8): Warning:  : SR0007 : Microsoft.Rules.Data : Nullable columns can cause final results to be evaluated as NULL for the predicate.
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(245,9): Warning:  : SR0007 : Microsoft.Rules.Data : Nullable columns can cause final results to be evaluated as NULL for the predicate.
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(301,9): Warning:  : SR0007 : Microsoft.Rules.Data : Nullable columns can cause final results to be evaluated as NULL for the predicate.
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(326,7): Warning:  : SR0007 : Microsoft.Rules.Data : Nullable columns can cause final results to be evaluated as NULL for the predicate.
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(475,8): Warning:  : SR0007 : Microsoft.Rules.Data : Nullable columns can cause final results to be evaluated as NULL for the predicate.
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(439,13): Warning:  : SR0007 : Microsoft.Rules.Data : Nullable columns can cause final results to be evaluated as NULL for the predicate.
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(439,29): Warning:  : SR0007 : Microsoft.Rules.Data : Nullable columns can cause final results to be evaluated as NULL for the predicate.
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(446,13): Warning:  : SR0007 : Microsoft.Rules.Data : Nullable columns can cause final results to be evaluated as NULL for the predicate.
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(446,29): Warning:  : SR0007 : Microsoft.Rules.Data : Nullable columns can cause final results to be evaluated as NULL for the predicate.
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(453,13): Warning:  : SR0007 : Microsoft.Rules.Data : Nullable columns can cause final results to be evaluated as NULL for the predicate.
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(453,29): Warning:  : SR0007 : Microsoft.Rules.Data : Nullable columns can cause final results to be evaluated as NULL for the predicate.
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(460,9): Warning:  : SR0007 : Microsoft.Rules.Data : Nullable columns can cause final results to be evaluated as NULL for the predicate.
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(518,9): Warning:  : SR0007 : Microsoft.Rules.Data : Nullable columns can cause final results to be evaluated as NULL for the predicate.
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(243,4): Warning:  : SR0014 : Microsoft.Rules.Data : Data loss might occur when casting from Money to Int.
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(326,34): Warning:  : SR0014 : Microsoft.Rules.Data : Data loss might occur when casting from Date to DateTime.
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(326,49): Warning:  : SR0014 : Microsoft.Rules.Data : Data loss might occur when casting from Date to DateTime.
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(390,60): Warning:  : SR0014 : Microsoft.Rules.Data : Data loss might occur when casting from Money to Decimal(3, 2).
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(475,35): Warning:  : SR0014 : Microsoft.Rules.Data : Data loss might occur when casting from Date to DateTime.
demo\Stored Procedures\sp_NightlyProcessingForReporting.sql(475,50): Warning:  : SR0014 : Microsoft.Rules.Data : Data loss might occur when casting from Date to DateTime.
```


```
: Microsoft.Rules.Data : Data loss might occur when casting from Money to Int.
4 : Microsoft.Rules.Data : Data loss might occur when casting from Date to DateTime.
4 : Microsoft.Rules.Data : Data loss might occur when casting from Date to DateTime.
4 : Microsoft.Rules.Data : Data loss might occur when casting from Money to Decimal(3, 2).
```


```
demo\Stored Procedures\sp_SearchAllSoldInventory.sql(2,25): Warning:  : SR0016 : Microsoft.Rules.Data : Stored procedure(sp_SearchAllSoldInventory) includes sp_ prefix in its name.
```

**The results are saved in \bin\Debug\<db project>.StaticCodeAnalysis.Results.xml**

# Dependencies - sp_helpExpandView

- Community tool by Andy Yun

```
26 /* run sp_helpExpandView for the main proc. Includes information for the related stored procedure */
27 EXEC sp_helpExpandView @ViewName = '[demo].[sp_NightlyProcessingForReporting]', @OutputFormat = 'horizontal'
28
```

sults Messages

| BaseObject_FullName | Lvl_1 | Obj_1 | Typ_1 | Lvl_2 | Obj_2 | Typ_2 | Lvl_3 | Obj_3 | Typ_3 | Lvl_4 | Obj_4 | Typ_4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| demo.sp_NightlyProcessingForReporting | 1 | NULL | NULL | 2 | | | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | dbo.Customer | U | 2 | | | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | dbo.Inventory | U | 2 | | | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | dbo.SalesHistory | U | 2 | | | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | dbo.SalesPerson | U | 2 | | | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | dbo.vw_salesperson_annualnumofsales | V | 2 | dbo.vw_SalesPerson_SalesPerMonth | V | 3 | dbo.BaseVw_SalesHistory | V | 4 | dbo.SalesHistory | U |
| demo.sp_NightlyProcessingForReporting | 1 | dbo.vw_salesperson_annualnumofsales | V | 2 | dbo.vw_SalesPerson_SalesPerMonth | V | 3 | dbo.BaseVw_SalesPerson | V | 4 | dbo.SalesPerson | U |
| demo.sp_NightlyProcessingForReporting | 1 | demo.NightlySalesStaging | U | 2 | | | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | demo.rpt_Avg_Profit_Classification_Type | U | 2 | | | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | demo.rpt_Avg_Vehicle_Age_Classification_Type | U | 2 | | | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | demo.sp_searchallsoldinventory | P | 2 | dbo.Inventory | U | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | demo.sp_searchallsoldinventory | P | 2 | dbo.SalesHistory | U | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | demo.sp_searchallsoldinventory | P | 2 | Vehicle.BaseModel | U | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | demo.sp_searchallsoldinventory | P | 2 | Vehicle.Color | U | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | demo.sp_searchallsoldinventory | P | 2 | Vehicle.Make | U | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | demo.sp_searchallsoldinventory | P | 2 | Vehicle.Model | U | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | demo.sp_searchallsoldinventory | P | 2 | Vehicle.Package | U | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | demo.udf_CalculateNetProfit | FN | 2 | demo.NightlySalesStaging | U | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | Vehicle.BaseModel | U | 2 | | | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | Vehicle.Classification | U | 2 | | | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | Vehicle.Color | U | 2 | | | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | Vehicle.Make | U | 2 | | | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | Vehicle.Model | U | 2 | | | 3 | | | 4 | | |
| demo.sp_NightlyProcessingForReporting | 1 | Vehicle.Package | U | 2 | | | 3 | | | 4 | | |

# Gather Performance Information

- spWhoIsActive
- Query Store
- I/O characteristics
- Extended Events
  - Run time stats
  - Other activity

# Gather Performance Information (cont'd)

- Table stats for objects used
  - Number of rows
    - MAX & AVG over clients data
    - Diff between client and dev data
  - Constraints\Indexes
  - Triggers
- 3rd Party Monitoring

# Determine the Fix

- Do the fixes have to be done together or separately?
- How much of the code needs to be changed?
- Are there different options for each fix?
- Which require changes outside of the code?
  - Table\Index changes
  - Application code changes

# Design Test Plans

- Types of Tests:
  - Performance impact
  - Unit testing
  - Bug fixes
  - Regression testing
- Are there environments to support testing?

# Compile the Results of the Technical Analysis

| Line # | Issue | Bug | Perf Concern | How to fix | How to test |
|---|---|---|---|---|---|
| 79 | comment says Merge statement has insert, update and delete; no delete statement | M | Y | if delete statement is supposed to be there, add it in | Add a record to demo.NightlySalesStaging that doesn't exist in dbo.SalesHistory.<br>Confirm it doesn't exist after proc is run |
| 172 | Drop Table and recreate through SELECT INTO | N | Y | change DROP TABLE to TRUNCATE TABLE<br>change SELECT INTO into INSERT INTO statement<br>confirm table definition is correct | confirm that stored procedure still works<br>confirm performance isn't impacted if indexes are added |
| 192 | Drop Table and recreate through SELECT INTO | N | Y | change DROP TABLE to TRUNCATE TABLE<br>change SELECT INTO into INSERT INTO statement<br>confirm table definition is correct | confirm that stored procedure still works<br>confirm performance isn't impacted if indexes are added |
| 212 | Drop Table and recreate through SELECT INTO (rpt_Avg_Profit_Classification_Type) | Y | Y | BUG: create table statements creates an integer. Populated by an scalar UDF that returns a money datatype.<br>change DROP TABLE to TRUNCATE TABLE<br>change SELECT INTO into INSERT INTO statement<br>confirm table definition is correct | confirm that stored procedure still works<br>confirm performance isn't impacted if indexes are added |
| 268 | Drop Table and recreate through SELECT INTO | N | Y | change DROP TABLE to TRUNCATE TABLE<br>change SELECT INTO into INSERT INTO statement<br>confirm table definition is correct | confirm that stored procedure still works<br>confirm performance isn't impacted if indexes are added |

# Business Plan of Attack

# Business Plan of Attack

- Prioritize the issues
- Estimate time
- Additional resource needs
- Development\Release schedules
- Prioritize with existing projects

# Prioritize the issues

- Which needs the least effort?
- Which are the riskiest?
- Which has the biggest application impact?
- Which has the biggest business impact?

# Estimate Time

- How long does each fix take?
- How does that fit in with existing development workflows?
  - Sprints
- What's the estimated time to test each fix?
  - Unit testing
  - Regression testing

# Determine Additional Resource Needs

- QA
  - Performance Impact testing
  - Regression testing
- Developers
  - Application changes
- Database developers
  - How many are available?

# Development & Release Schedules

- Release planning
  - General release
  - Hot fix
  - Client one-off hot fix

# Prioritize with Existing Projects

- How will the changes affect other project timelines?
- Should this project take priority over other projects?
- How to convince project managers?

# Bringing it all together

Make your business case

# Final Thoughts...

# Perfect is the enemy of good

3 languages

Article Talk Tools

From Wikipedia, the free encyclopedia

**Perfect is the enemy of good** is an aphorism which means insistence on perfection often p… 80–20 rule ex… me to comple… he effort.[1… effort results in diminishing returns, further activity becomes increasingly inefficient.

The first 80% takes 20% of the time.

The last 20% takes 80% of the time.



https://en.wikipedia.org/wiki/Perfect_is_the_enemy_of_good

# Additional Resources

- Redgate: "How Do My Peers Do This?" The latest best practices for IT architects implementing a major business initiative

# Additional Resources - Tools

- Code Analysis:
  - Mala Mahadevan: Stairway to ScriptDOM
  - Redgate: SQL Prompt SQL Code Analysis
  - Microsoft: Overview of Extensibility For Database Code Analysis Rules

# Additional Resources – Tools (cont'd)

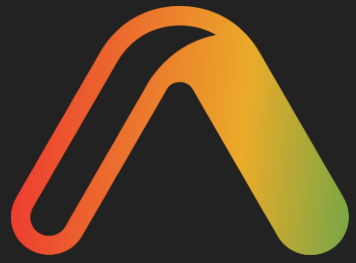
- sp_helpExpandView (Andy Yun)
- sp_HumanEvents (Erik Darling)
- sp_QuickieStore (Erik Darling)

# Session evaluation

Your feedback is important to us

**Evaluate this session at:**

www.PASSDataCommunitySummit.com/evaluation



PASS
Data Community
SUMMIT
2023

# Thank you

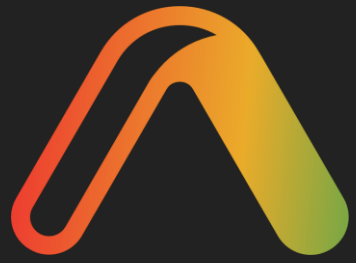
If you have any other questions, feel free to reach out and ask!

**Deborah Melkin**

Email: dgmelkin@gmail.com

Blog: DebtheDBA.wordpress.com

GitHub: github.com/DebTheDBA



PASS
Data Community
SUMMIT
2023