

Refactoring Monolith Database Stored Procedures

@RileyMajor



Thanks to our sponsors

Quest

⋮ Profisee

DELL EMC

cozyroc

SIOS

vmware

SentryOne®

TALAVANT

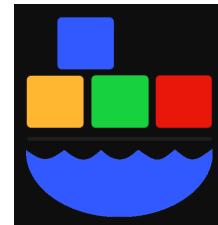
PURE STORAGE®

ProcureSQL

Microsoft Azure

Insight
Digital Innovation

IDERA



redgate



Monowhat?

- Monolith
 - Usually evolves over time.
 - Long.
 - Does multiple things.
 - Often repetitious.
 - Often disorganized.
 - “Big ball of mud.”
 - Fragile.
 - Untestable.
 - Scary.

Layers. Like an onion.

- Programming Layers
 - Presentation
 - Business Logic
 - Data Storage
- Tiers
 - Client Application (JS / HTML or Binary)
 - Server (PHP / NodeJS / ASP.NET)
 - Database (SQL Server, MySQL, NoSQL)
- “Best Practice”
 - Presentation in Client Application
 - Business Logic in Server

Oh noes!

- Monolith
 - Presentation in Database!
 - Business Logic in Database!
- Bad
 - Database scaling is \$\$\$hard\$\$\$.
 - Causes vendor lock-in.
 - Database languages are primitive.
- But
 - Close to data. Less overhead.
 - Who really changes databases?
 - SQL more powerful than you think.

It's turtles all the way down.

- Separate layers even in the same tier.
 - Browser: MVVM.
 - Server: MVC.
 - Database: TBD.
- Database Layer
 - Presentation / Business Logic
 - IF / CASE / SET / SUM / + / DATEADD
 - Data Access
 - SELECT / UPDATE / INSERT / DELETE
- Testability, Isolation, and Portability.

Make a plan.

- What are your goals?
 - Better performance?
 - Easier to maintain?
 - Easier to understand?
 - Easier to test?
- How will you know you've achieved them?
 - Speed benchmarks?
 - Less repetition?
 - Smaller sections of code?
 - Actually having a testing suite?

Survey the damage.

- You can't avoid a thorough code review.
- Look for data modification.
 - INSERT, UPDATE, and DELETE.
 - Note columns affected.
- Look for external effects.
 - CLR
 - Email generation.
- Look for transaction handling.
 - BEGIN TRAN, COMMIT TRAN, ROLLBACK TRAN

Don't break anything

- Build a development environment.
 - You need to be able to play around.
 - You need realistic data (volume and content).
 - But maybe not real data.
- Work in isolation.
 - Were the changes from your process or another?
 - Is it slow because resources are used elsewhere?
- How can you tell if you broke something?
 - You need to capture a before and after state.
 - Aim for a deterministic process.

Deterministic

- Function always returns the same value for the same inputs.
- Easy to test: send the same values in before and after changes.

Yes	No	No	No
Return $x + y$	Return $x + y +$ rand()	Return $x + y +$ hour(now())	Return $x + y +$ (SELECT TOP 1 z FROM Table)

Play it again, Sam.

- Good
 - $X \rightarrow Y$
 - :: changes ::
 - $X \rightarrow Y$
- Bad
 - $X \rightarrow Y$
 - :: changes ::
 - $X \rightarrow Z$

Good luck with that.

- Real world not deterministic.
- Monoliths change state.
- Running second time not the same.
- $\neg \exists (\forall)$
 - $X \rightarrow Y$
 - :: changes ::
 - $Y \rightarrow Q$
- Need to go back in time.
- Transactions!



Become a wrapper.

- To test impact of code changes, wrap your calls:
 - Begin transaction.
 - Run original code.
 - Capture changed data.
 - Rollback transaction (to revert data).
 - Run new code.
 - Capture changed data.
- Now compare the 2 captured data sets...

Oops

- Where did the changes go?
- Captured data is also rolled back!
- How can you save data which has been killed?

Build a Ghost House

- How capture doomed data?
 - Outside SQL Server? Hard.
 - Another thread with NOLOCK? Hard.
- What's immune from transactions?
- Variables!
- You can't have a variable for every row.
- One big XML? Ouch.
- Table variables survive transactions.
- They're ghost houses!

Spooky Playground - Create House

```
DECLARE @Orders TABLE  
(  
    CompareOrderID int,  
    ColA_Orig int,  
    ColA_AfterMonolith int,  
    ColA_AfterChanges int  
);
```

Spooky Playground - Fill House

```
INSERT INTO @Orders
(
    CompareOrderID, ColA_Orig
)
SELECT
    OrderID, ColA
FROM
    Orders;
```

Spooky Playground - Catch Ghosts

```
BEGIN TRAN; EXEC Monolith;
```

```
UPDATE @Orders  
SET ColA_AfterMonolith = ColA  
FROM Orders  
WHERE OrderID = CompareOrderID;
```

```
ROLLBACK TRAN;
```

Spooky Playground - Again

```
BEGIN TRAN; EXEC Monolith_New;
```

```
UPDATE @Orders  
SET ColA_AfterChanges = ColA  
FROM Orders  
WHERE OrderID = CompareOrderID;
```

```
ROLLBACK TRAN;
```

Spooky Playground - Compare

```
SELECT
  *
FROM
  @Orders
WHERE
  ColA_AfterMonolith <> ColA_AfterChanges;
```

Mock your Black Boxes

- Transactions only work on the database.
- External effects aren't rolled back.
- Replace external calls with "mocks".
- They look and act like external calls.
- But you control the guts.
- Return hard-coded sample data.
- Have the mock log its inputs.

Make your time!

- Date/Time functions kill determinism.
- You have to control "now".
- Otherwise no two runs could be the same.
- So make your own time.
- And send it in as a parameter.

Your petard should hoist.

- Move variable DECLAREs to the top.
- Reveals duplication.
- Reveals common data sources.
- Displays breadth of data required.
- Caution: DECLARE assignment.

One SELECT to rule them all.

- Gather scattered SELECT statements at top.
- Reveals duplication.
- Prepares for separation.
- Prepares for shorter transactions.
- Use a single SELECT with fancy SQL if practical.

Measure twice, cut once.

- Find INSERT/UPDATE/DELETE.
- Replace with variables SETs.
- Move data modification to end of proc.
- Results in 3 main sections:
 - Data gathering.
 - Computation
 - Data modification.

Cases of CASES

- What's left? Logic.
- Lots of IFs, SETs, and calculations.
- Pull it all together in one giant statement.
- Usually performs better.
- Can be clearer.
- Can reduce code.
- Prepares for separation.
- CASE, Derived Tables, and CTEs are your friends.



Building Blocks

- Still one procedure = Still a monolith.
- Separate
 - Data Gathering -> Inline UDFs
 - Calculation -> Inline UDF.
- Allows data gathering re-use.
- Allows testing suite for business rules.
- Allows read-only monolith actions.

It's all better now...

Monolith

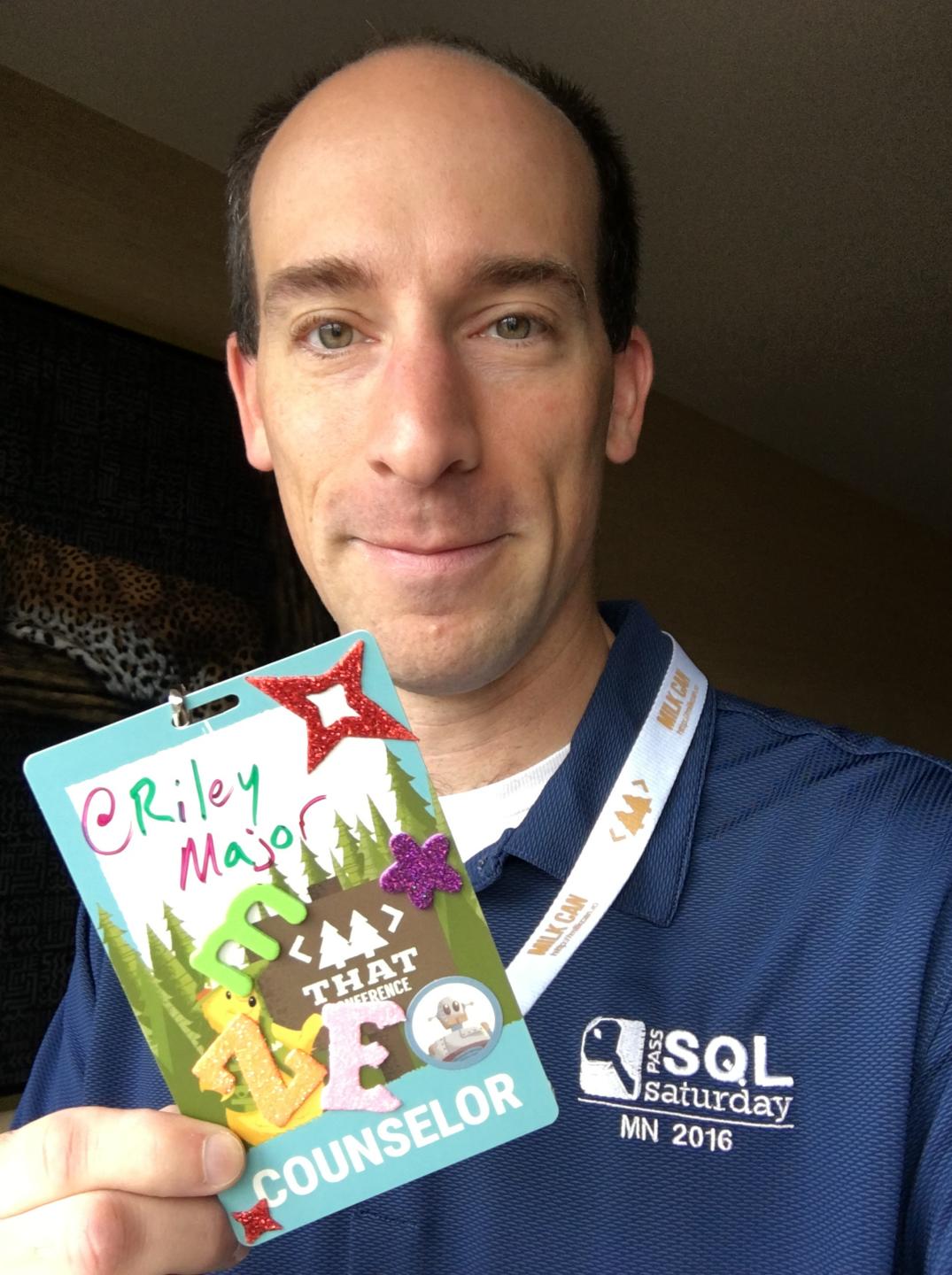
- Usually evolves over time.
- Long.
- Does multiple things.
- Often repetitious.
- Often disorganized.
- “Big ball of mud.”
- Fragile.
- Untestable.
- Scary.

Reformed monolith.

- Recently written.
- Short.
- Orchestrates multiple things.
- Repeated code eliminated.
- Organized into functions.
- Vials of reagents to mix.
- Problems isolated.
- Testable.
- Benign.

Riley Wheeler Major

PASSMN@RileyMajor.com
twitter.com/rileymajor
github.com/rileymajor
linkedin.com/in/rileymajor/



Enterprise Architect
Pilot Freight Services

Image Credits

Title	To-Do	Become a wrapper	One SELECT to rule them all
Sam Howzit Follow Campfire https://www.flickr.com/photos/aloha75/8608867490 8608867490_303df542c0_k.jpg	Nikki Buitendijk To do https://www.flickr.com/photos/nikkibuitendijk/14670721802 14670721802_4a6e2b9333_o.jpg	Matt Reinbold Hold All My Calls https://www.flickr.com/photos/furryscalyman/1034889957 1034889957_de68410cfa_o_crop.jpg	idreamlikecrazy One Ring to rule them all https://www.flickr.com/photos/purple-lover/13583362554 13583362554_33a81b110e_o_flipped.jpg
Monolith	Survey the Damage	Build a Ghost House	Measure Twice Cut Once
Institute for the Study of the Ancient World Follow The Mausoleum of Ateban (III) https://www.flickr.com/photos/isawnyu/5834281724/ 5834281724_99511f0d3a_o.jpg	Metropolitan Transportation Authority of the State of New York Follow Fix&Fortify - Greenpoint Tubes https://www.flickr.com/photos/mtaphotos/9504656210 9504656210_a7972b1039_o.jpg	Jamiecat * Amityville Haunted House https://www.flickr.com/photos/jamiecat/4910864986 4910864986_b1f81032a3_o.jpg	carrotmadman6 Scissors https://www.flickr.com/photos/carrotmadman6/4732395647 4732395647_e62cbe959c_o.jpg
Layers	Don't Break Anything	Spooky Playground	Cases of CASEs
John Fowler Layers https://www.flickr.com/photos/snowpeak/12547422744/ 12547422744_77bf3b1337_k.jpg	Stig Nygaard Trekkroner https://www.flickr.com/photos/stignygaard/34673211491 34673211491_ecfb9db477_o.jpg	Taber Andrew Bain Spooky Playground Equipment Detail https://www.flickr.com/photos/andrewbain/1542241604 1542241604_24f9cd70c7_o_crop.jpg	Lisa Parker crates https://www.flickr.com/photos/lisa-parker/4787352735 4787352735_32b926d54c_o_cropped.jpg
Oh Noes	Play it again, Sam!	Mock your black boxes	Building Blocks
Henry Burrows Surprised? https://www.flickr.com/photos/foilman/4998878794 4998878794_f554510315_o_crop.jpg	naql Record https://www.flickr.com/photos/naql/34691183380 34691183380_429a0c7120_k_crop.jpg	r2hox BLACK BOX - 6 https://www.flickr.com/photos/rh2ox/13890877727 13890877727_7d75107f1e_o_crop.jpg	Eric Kilby Augusta Block https://www.flickr.com/photos/ekilby/18099902934 18099902934_a12d573729_k.jpg
Turtles	Good luck with that.	Your Petard Should Hoist	All Better Now
andreistroe Stack of turtles https://www.flickr.com/photos/30312936@N04 4645520534_cbd2dcbae7_o_crop.jpg	JD Hancock Mad Man With A Box https://www.flickr.com/photos/jdhancock/7995637778 7995637778_75df99c01e_o.jpg	Larry Hoists https://www.flickr.com/photos/kayarewhy/14646253804 14646253804_035cbcc5aa_o.jpg	Waldemar Merger Green Hill https://www.flickr.com/photos/paxx/9546367406 9546367406_bcbe43bfe7_o.jpg