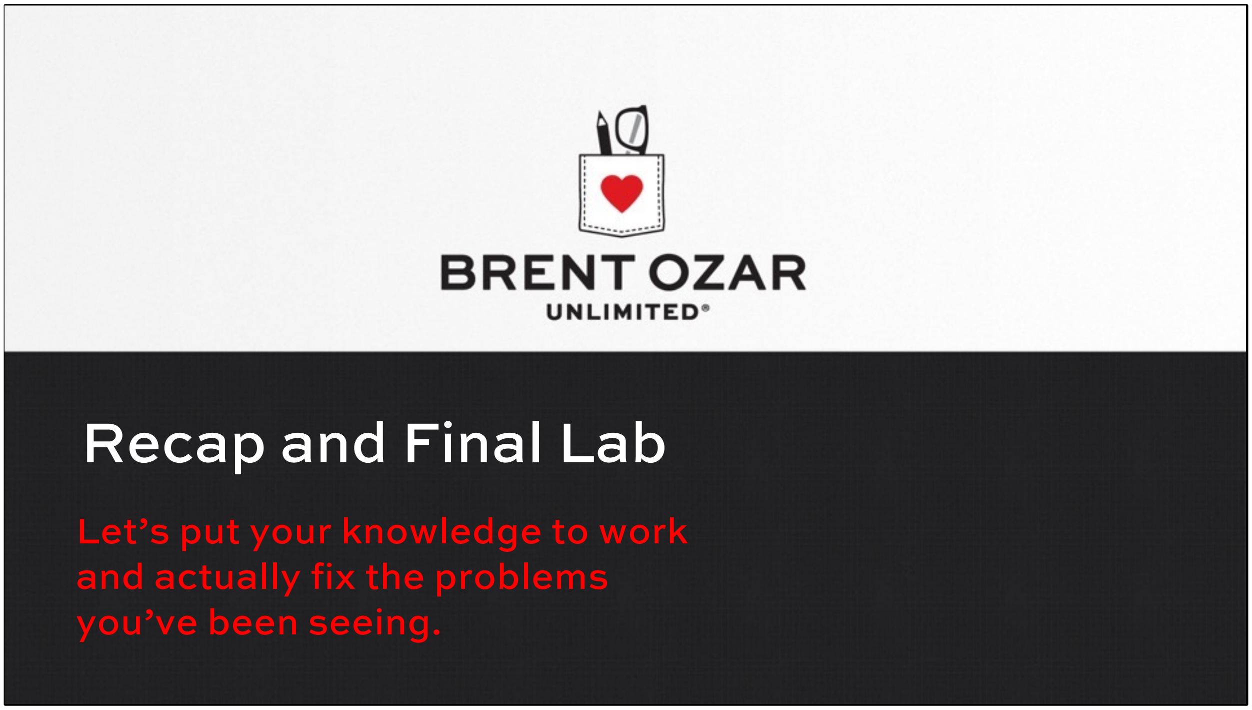
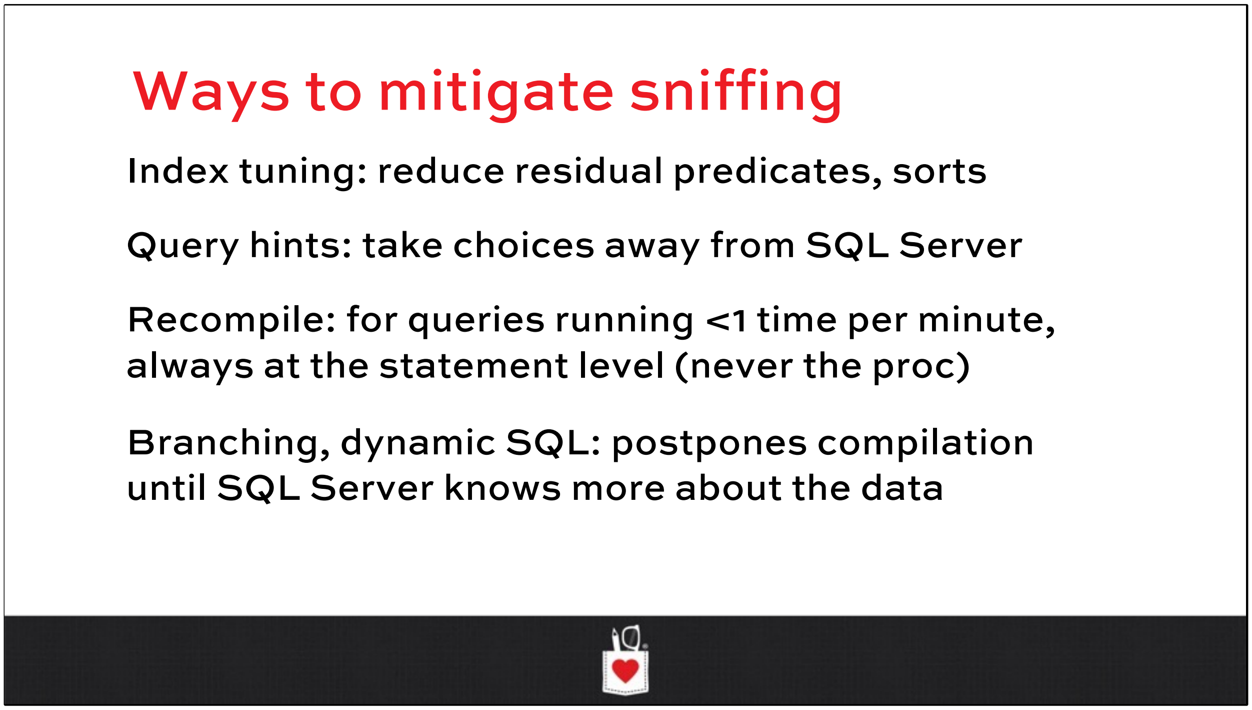


BRENT OZAR
UNLIMITED®

Recap and Final Lab

Let's put your knowledge to work and actually fix the problems you've been seeing.



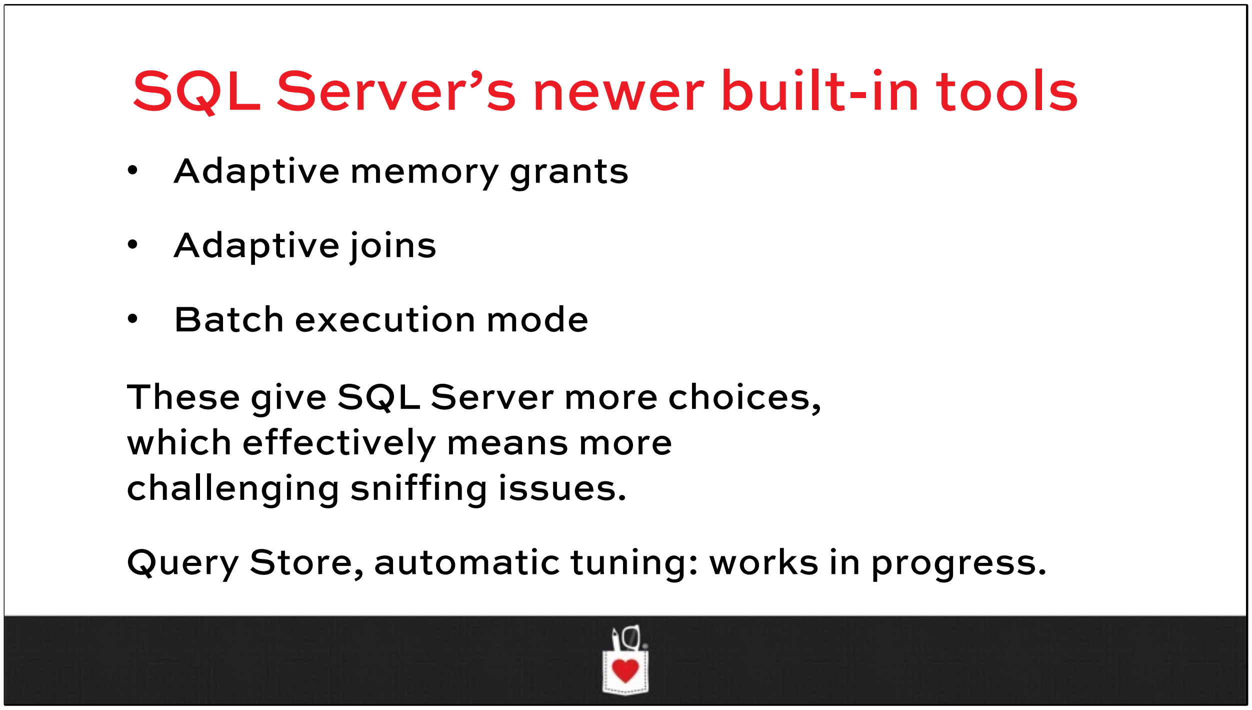
Ways to mitigate sniffing

Index tuning: reduce residual predicates, sorts

Query hints: take choices away from SQL Server

Recompile: for queries running <1 time per minute, always at the statement level (never the proc)

Branching, dynamic SQL: postpones compilation until SQL Server knows more about the data

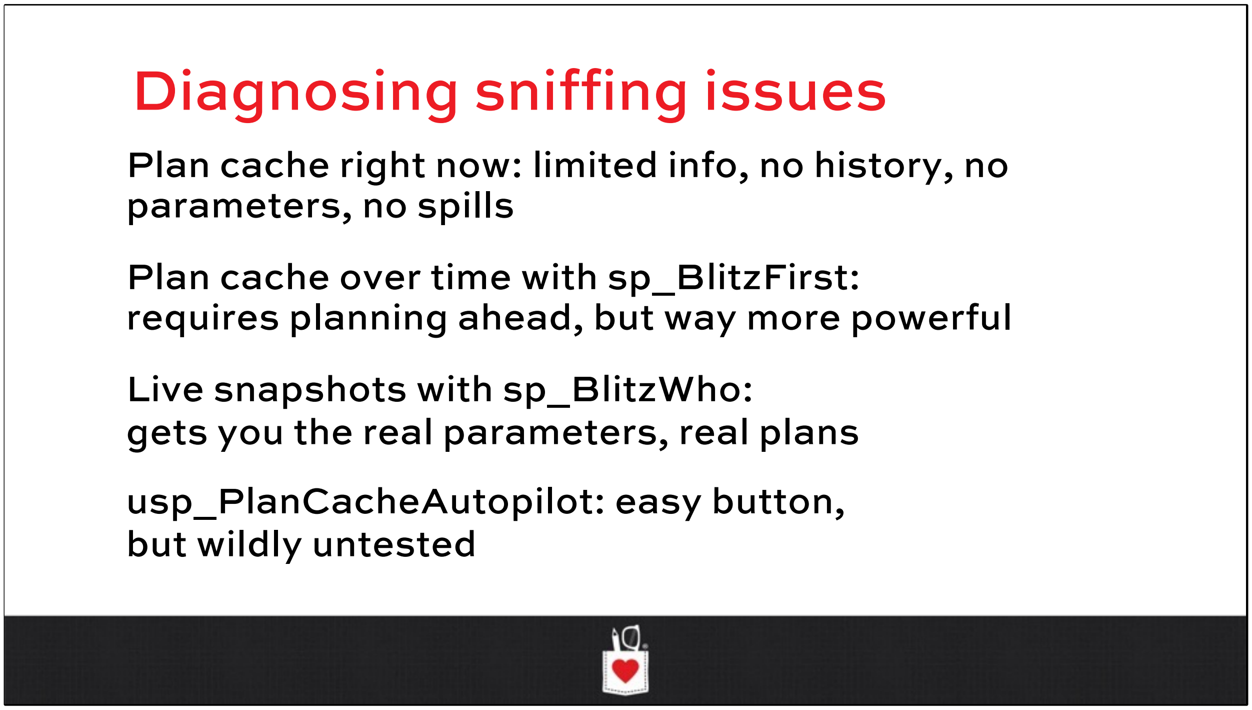


SQL Server's newer built-in tools

- Adaptive memory grants
- Adaptive joins
- Batch execution mode

These give SQL Server more choices,
which effectively means more
challenging sniffing issues.

Query Store, automatic tuning: works in progress.



Diagnosing sniffing issues

Plan cache right now: limited info, no history, no parameters, no spills

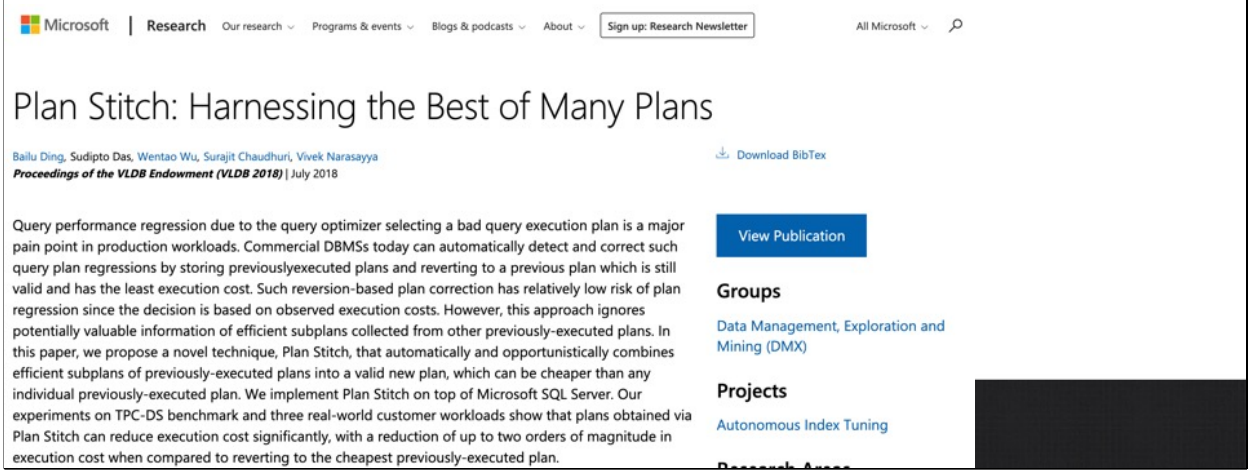
Plan cache over time with sp_BlitzFirst:
requires planning ahead, but way more powerful

Live snapshots with sp_BlitzWho:
gets you the real parameters, real plans

usp_PlanCacheAutopilot: easy button,
but wildly untested

The future we didn’t get

https://www.brentozar.com/archive/2019/08/research-paper-week-plan-stitch-harnessing-the-best-of-many-plans/



Plan Stitch: Harnessing the Best of Many Plans

Bailu Ding, Sudipto Das, Wentao Wu, Surajit Chaudhuri, Vivek Narasayya

Proceedings of the VLDB Endowment (VLDB 2018) | July 2018

Query performance regression due to the query optimizer selecting a bad query execution plan is a major pain point in production workloads. Commercial DBMSs today can automatically detect and correct such query plan regressions by storing previouslyexecuted plans and reverting to a previous plan which is still valid and has the least execution cost. Such reversion-based plan correction has relatively low risk of plan regression since the decision is based on observed execution costs. However, this approach ignores potentially valuable information of efficient subplans collected from other previously-executed plans. In this paper, we propose a novel technique, Plan Stitch, that automatically and opportunistically combines efficient subplans of previously-executed plans into a valid new plan, which can be cheaper than any individual previously-executed plan. We implement Plan Stitch on top of Microsoft SQL Server. Our experiments on TPC-DS benchmark and three real-world customer workloads show that plans obtained via Plan Stitch can reduce execution cost significantly, with a reduction of up to two orders of magnitude in execution cost when compared to reverting to the cheapest previously-executed plan.

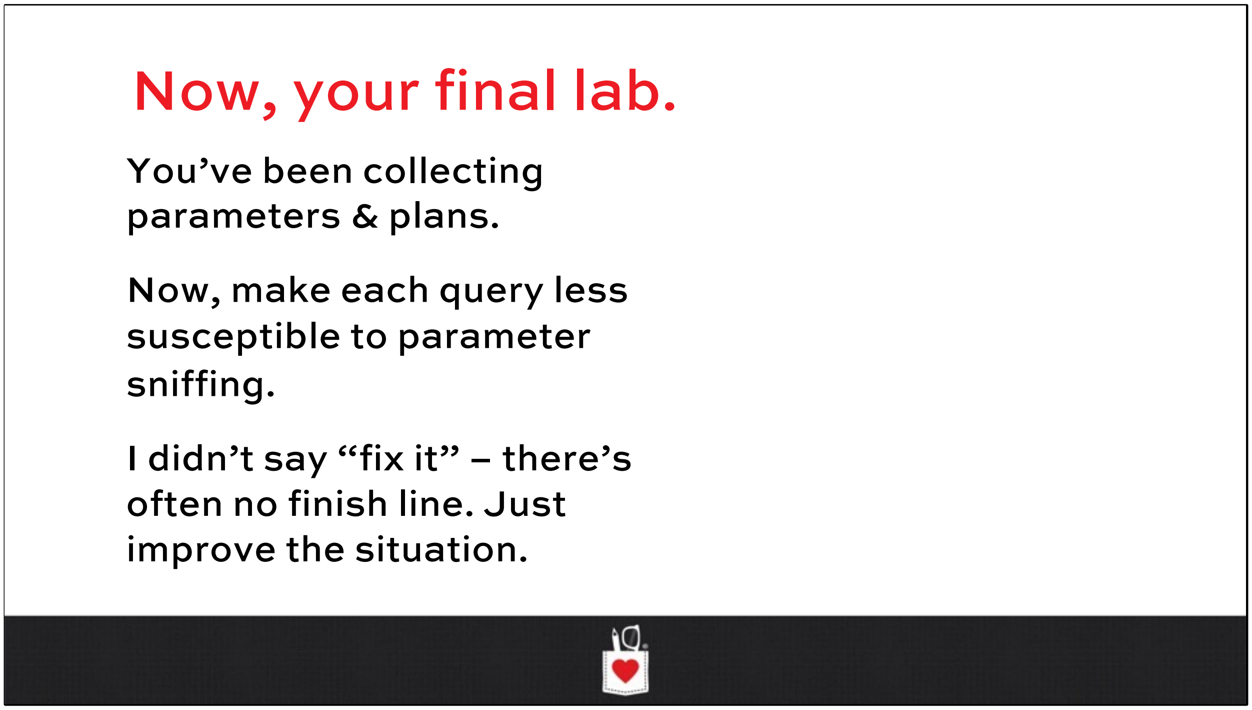
Groups

Data Management, Exploration and Mining (DMX)

Projects

Autonomous Index Tuning

Research Areas

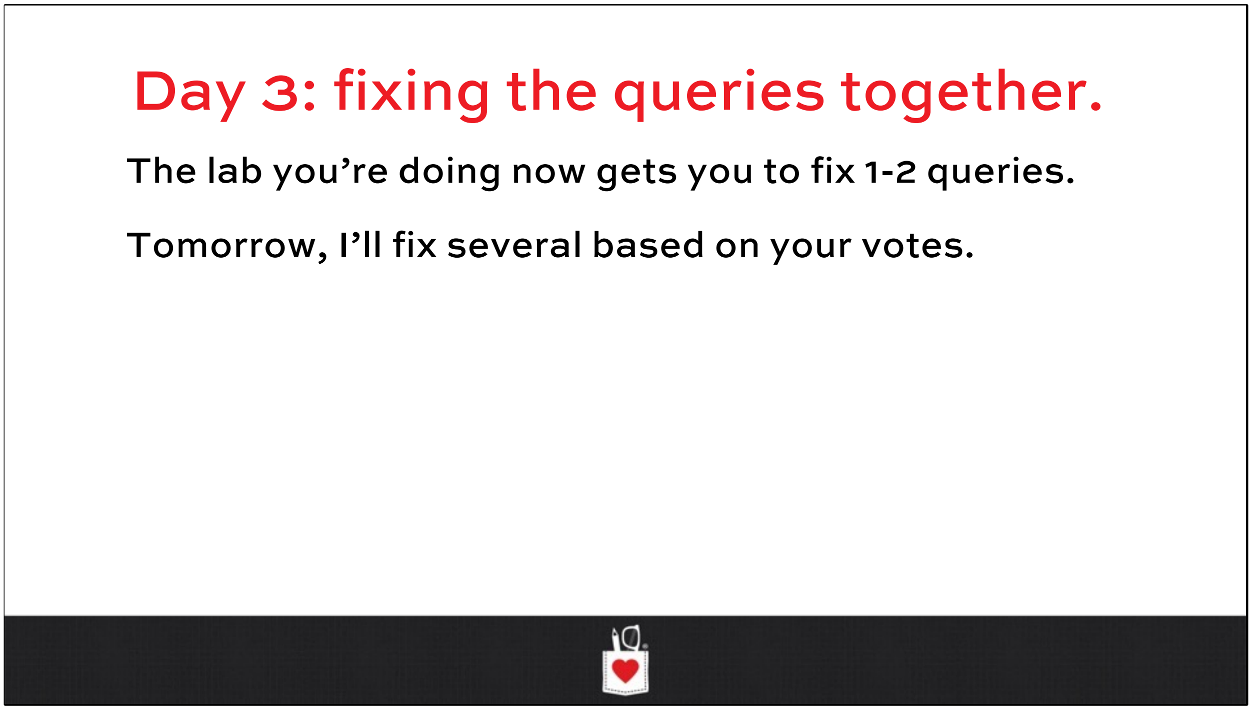


Now, your final lab.

You've been collecting parameters & plans.

Now, make each query less susceptible to parameter sniffing.

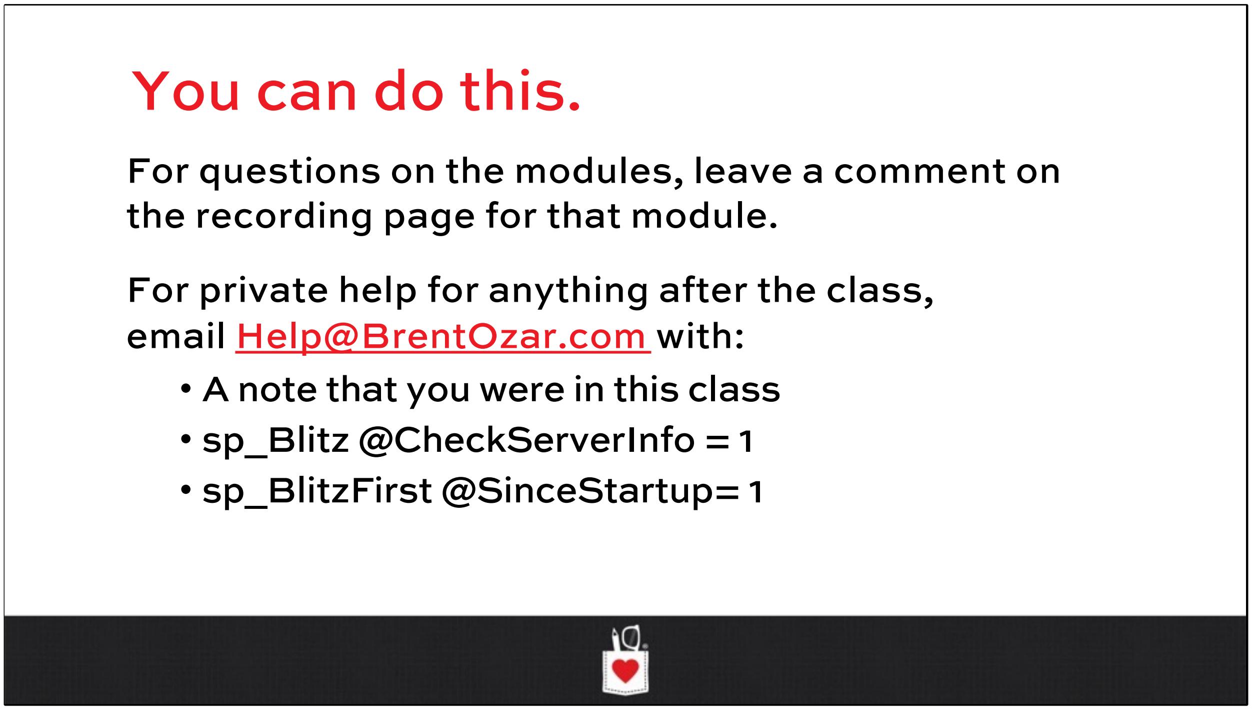
I didn't say "fix it" – there's often no finish line. Just improve the situation.



Day 3: fixing the queries together.

The lab you’re doing now gets you to fix 1-2 queries.

Tomorrow, I’ll fix several based on your votes.



You can do this.

For questions on the modules, leave a comment on the recording page for that module.

For private help for anything after the class, email Help@BrentOzar.com with:

- A note that you were in this class
- sp_Blitz @CheckServerInfo = 1
- sp_BlitzFirst @SinceStartup= 1