

Preventing SQL Server Performance Problems before they hit Production

Mark Allison

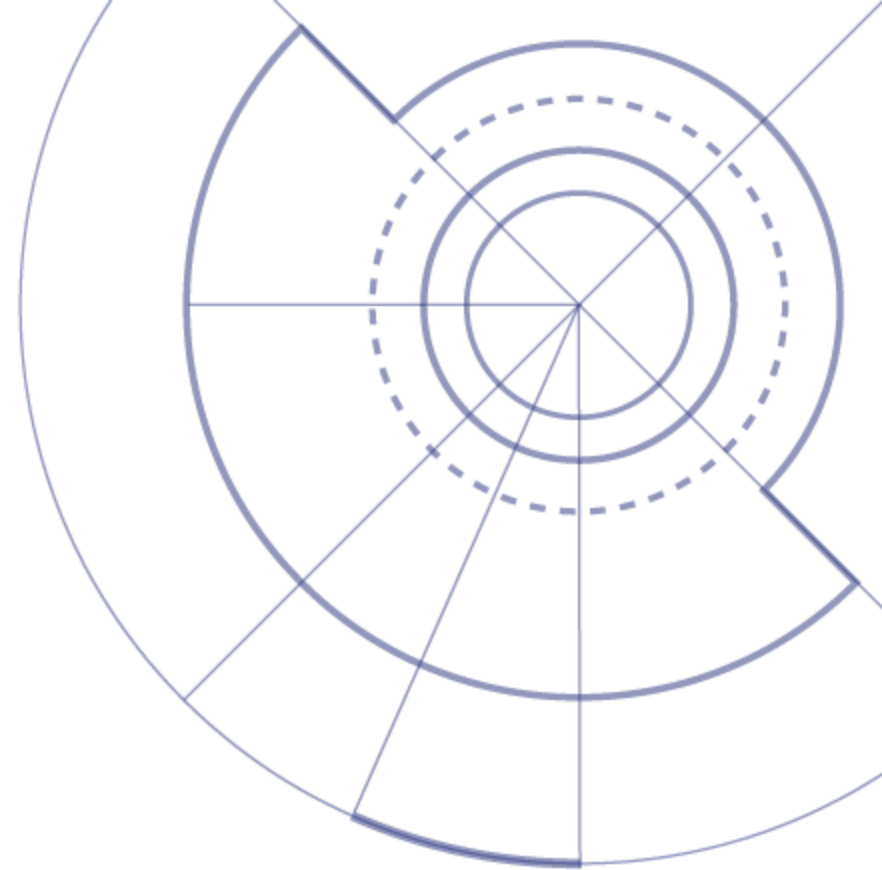
mark.allison@sabin.io

@dataguzzle

Paul Anderton

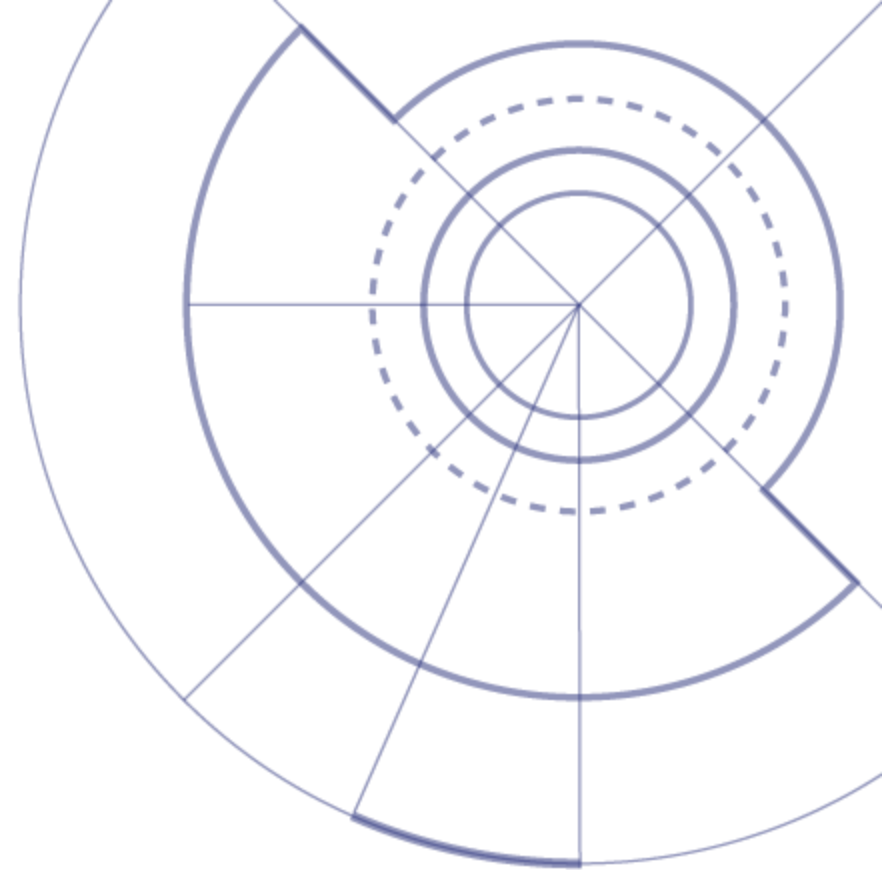
paul.anderton@sabin.io

@sqlpanders



Scenario:

A breaking change has been made



Who are we?

Mark Allison

- Worked with SQL Server since 1996
- Received MVP award 2002-2006
- Mainly investment banking



Who are we?

Paul Anderton

- Worked with SQL Server since 2000
- MCM: SQL Server
- Runs SQL Server User Group: Manchester, UK
- SQL Saturday Manchester, UK



Who are Sabin.io?

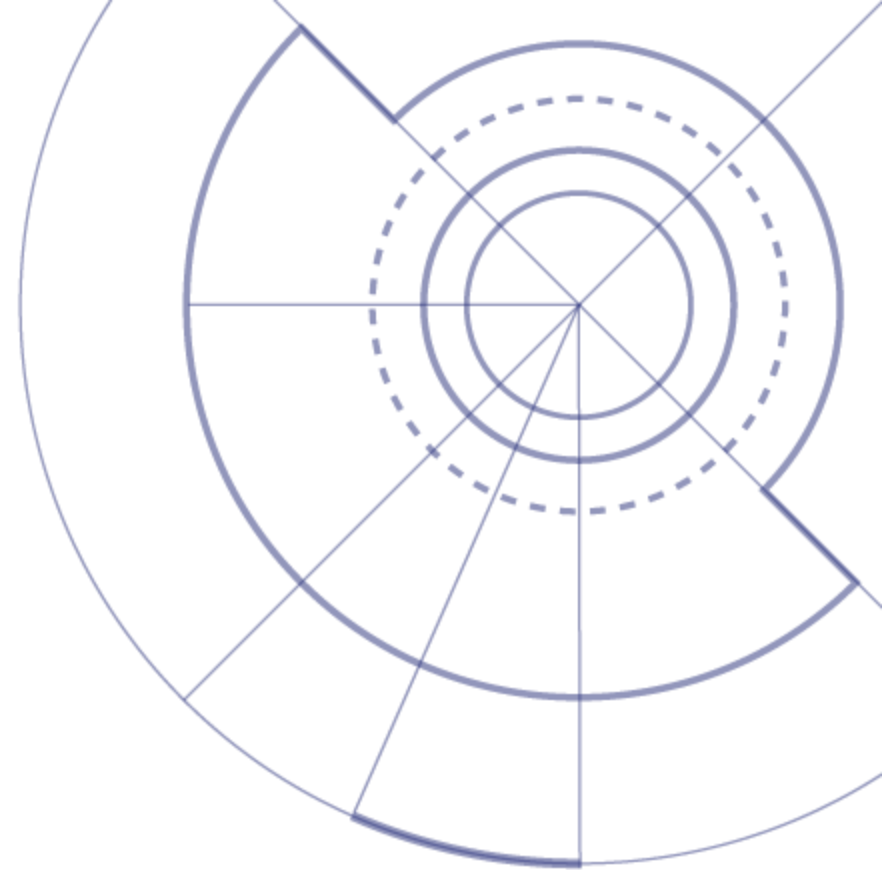
Founded by Simon Sabin

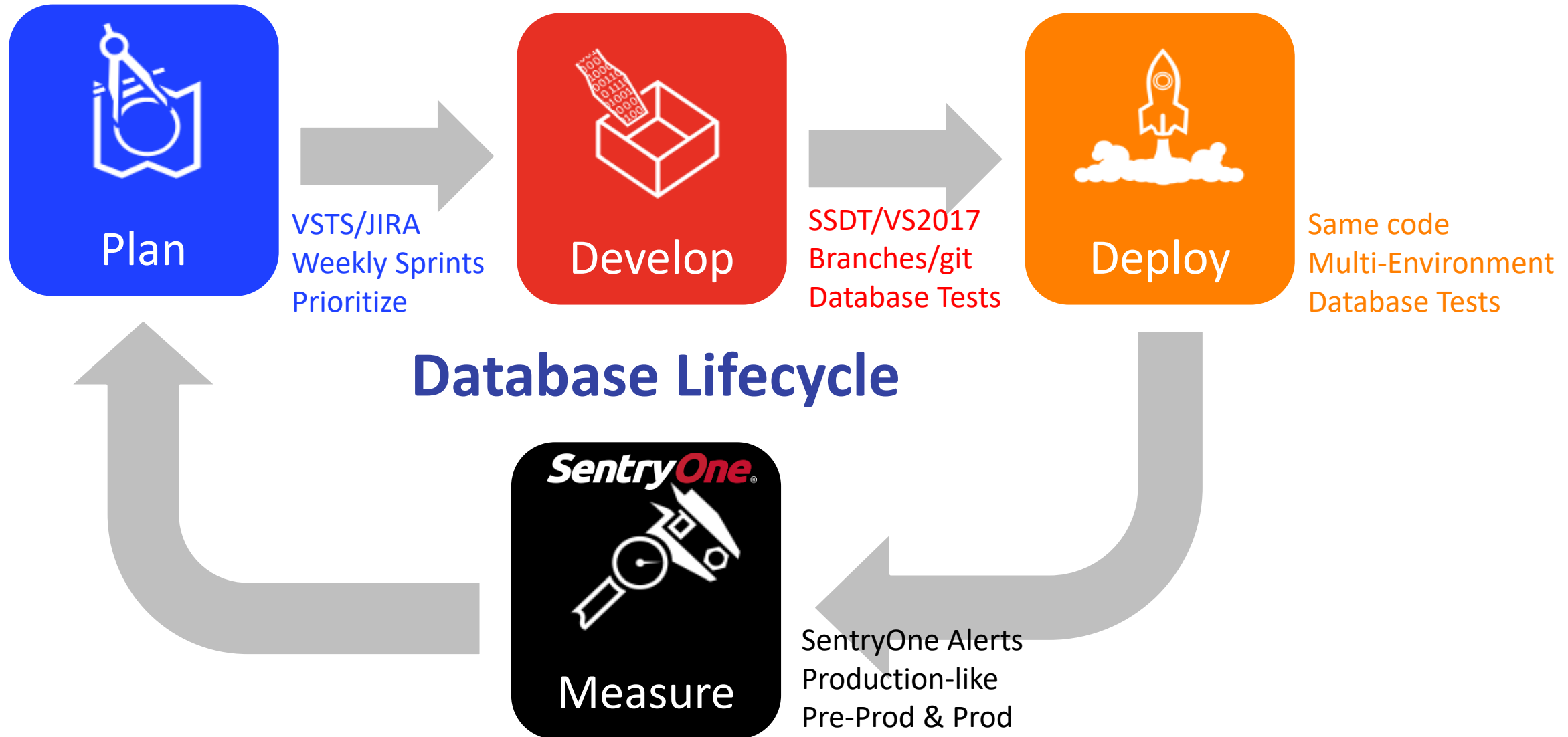
- MCM: SQL Server
- MVP & Microsoft Regional Director
- Founder of **sqlbits**
- Passionate about Data DevOps



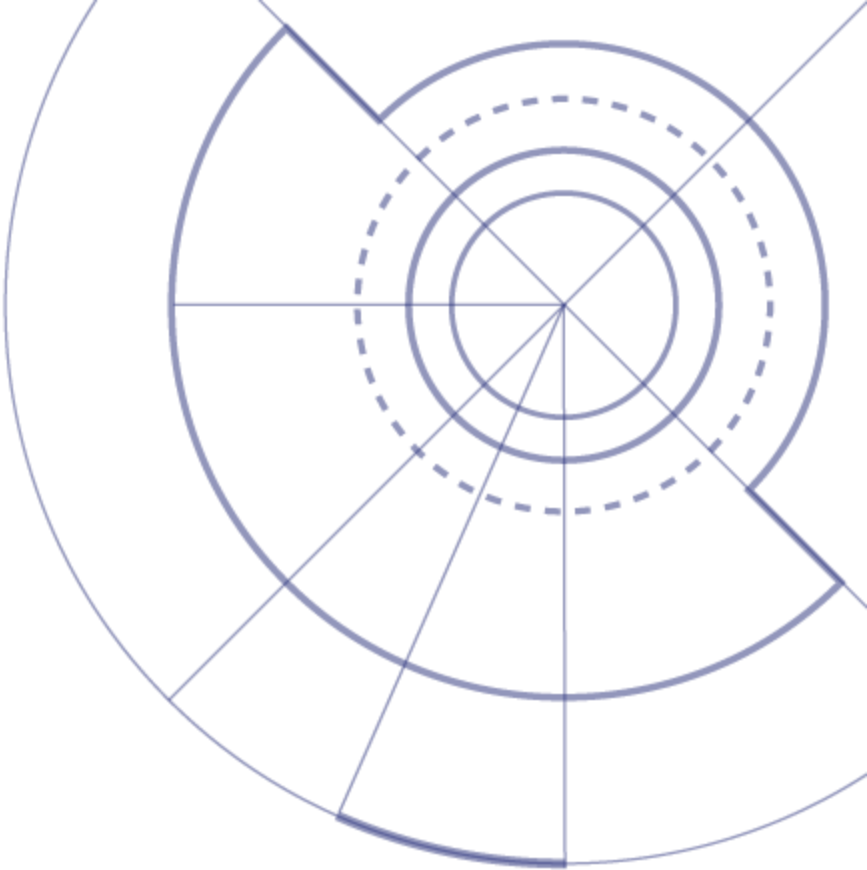
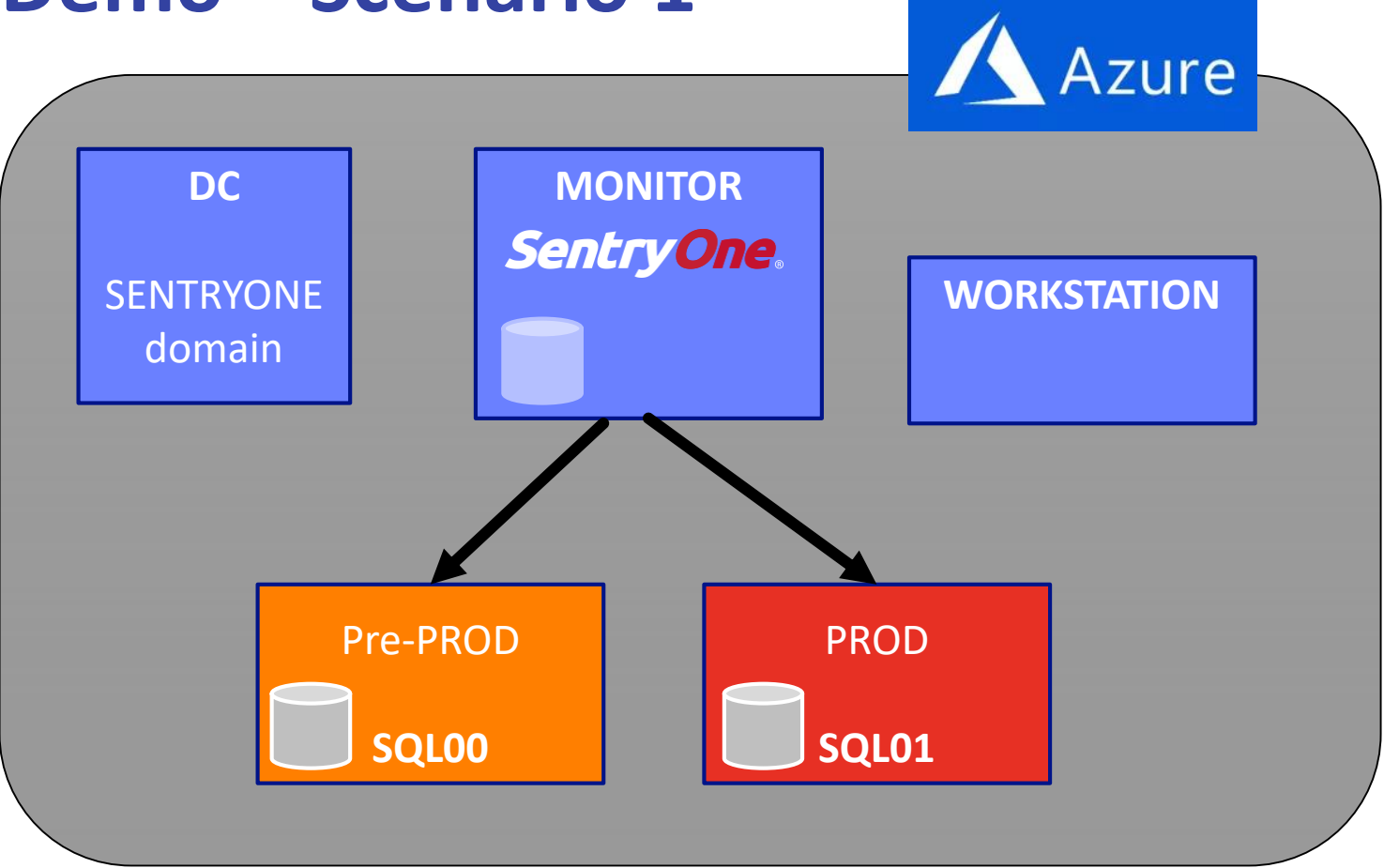
What are we covering today?

- How to deliver database change
- Demo:
 - One change resulting in excessive key lookups
 - One change resulting in excessive deadlocks
- Correlate in Sentry One
- Platform: Azure Windows 2016 VMs, SQL Server 2016, VSTS



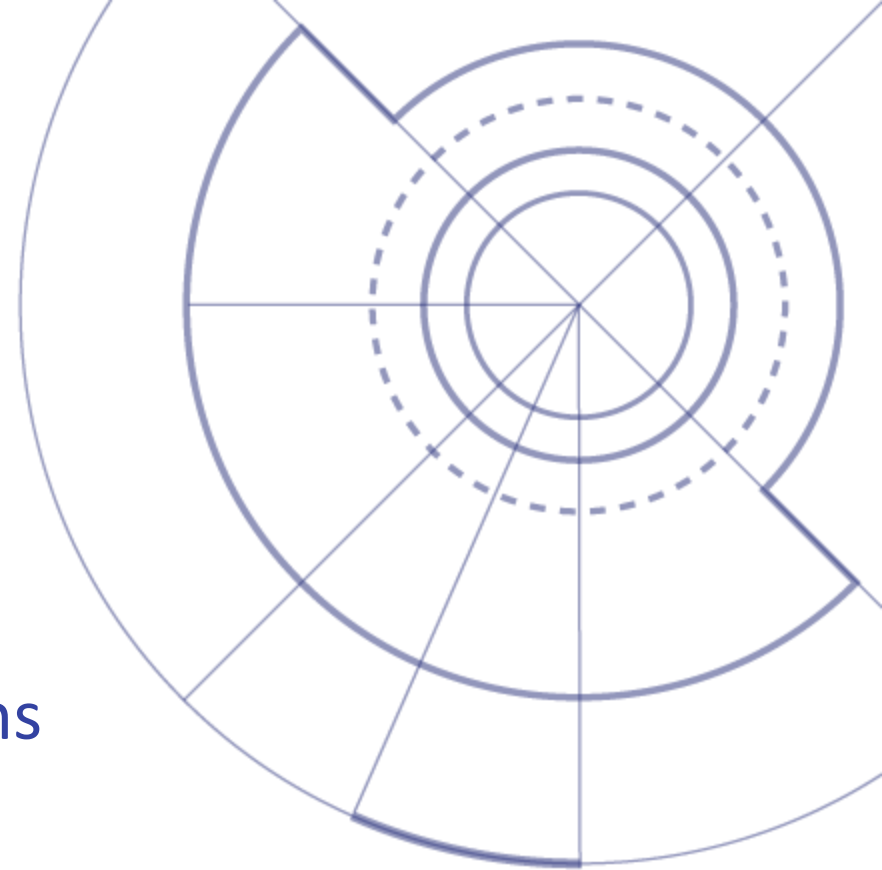


Demo – Scenario 1



Excessive Key Lookups Recap

- What is a Key Lookup?
- You have an index seek, but query requires columns not in index
- Solve with covering index, with **INCLUDE** columns
- Included columns are at leaf level
 - Columns in **SELECT** list that are not in **WHERE** clause
- Smaller faster indexes than covering indexes



Excessive Key Lookups - Example

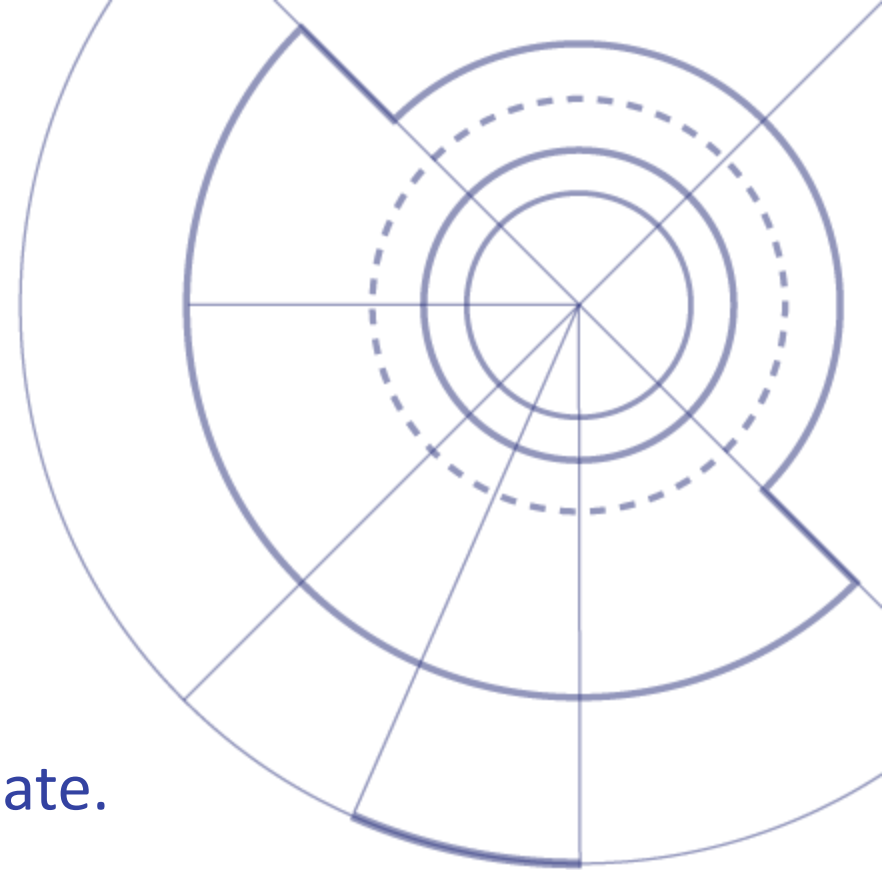
- For example, this query

```
SELECT BasketId, CustomerId, OrderDate  
FROM Sales.Basket
```

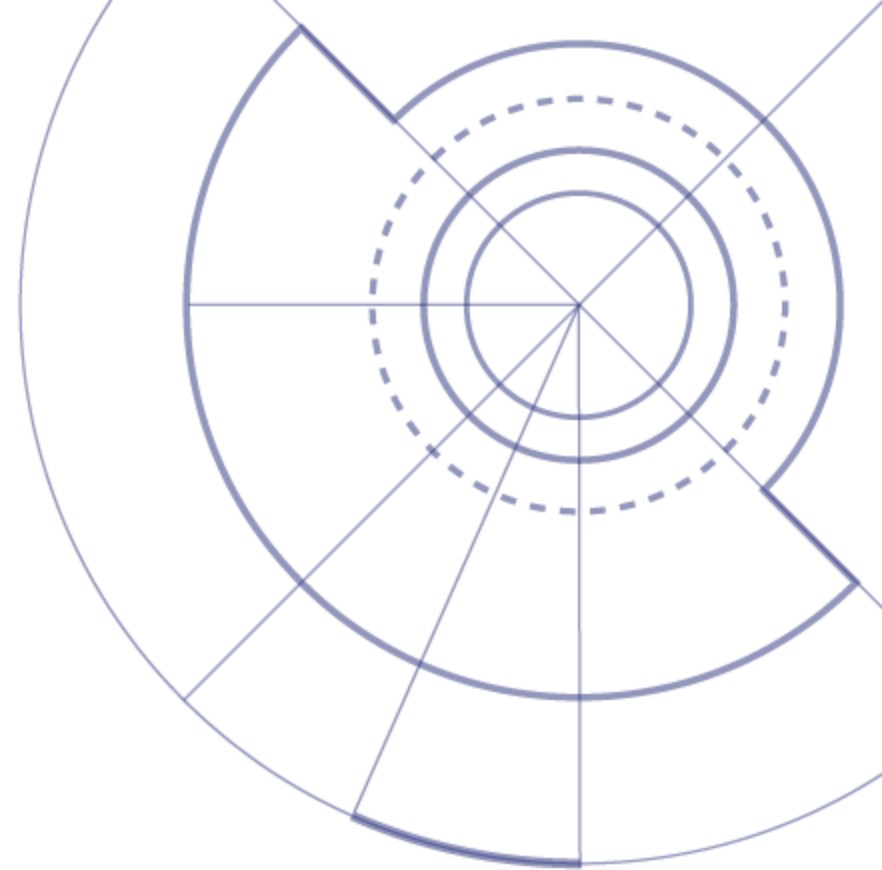
```
WHERE CustomerId = @CustomerId
```

- Index col on CustomerId, INCLUDE BasketId, OrderDate.

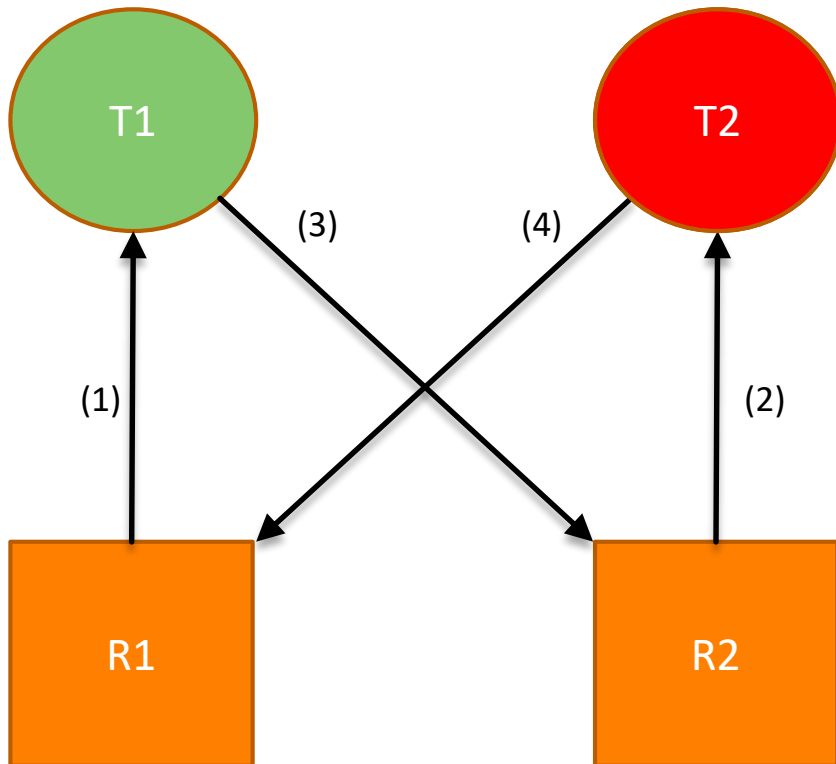
```
CREATE NONCLUSTERED INDEX [MyIndex]  
ON [Sales].[Basket] ([CustomerId])  
INCLUDE ([BasketID],[OrderDate])
```



Demo - Scenario 2



What is a Deadlock?



- Task 1 has a lock on Resource 1
- Task 2 has a lock on Resource 2
- Task 1 requests a lock on Resource 2
- Task 2 requests a lock on Resource 1
- Deadlock Monitor kills Task 2

Overview - Lifecycle

- **Plan** – agile methodology, work to measured sprints
- **Develop** – commit code to source control, make use of branches
- **Deploy** – database code committed should auto build and deploy to a build environment
- **Verify** – database tests using Pester, tSQLt, C#, whatever.
- **Measure** – release tested code to your environments. Use SentryOne for Perf-Test and Production environments

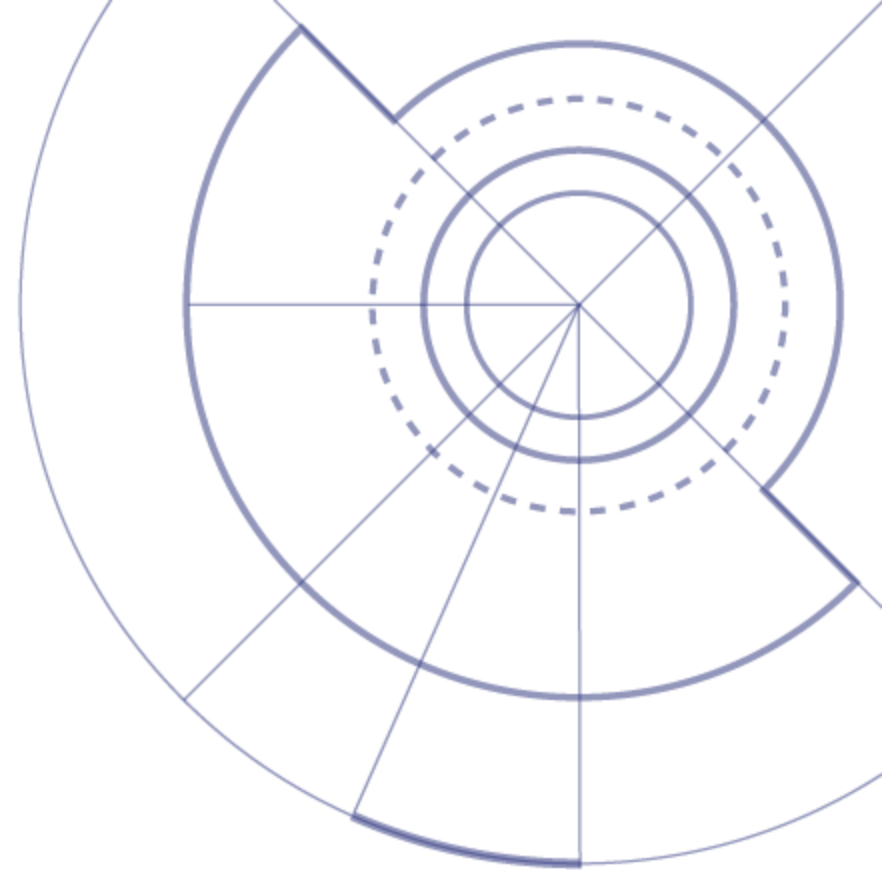
Summary

Advisory Conditions help

- Identify problems
- Surface external events

SentryOne

- Correlate multiple events
- Find problems quickly
- Drill into detail



Resources

Download Demo Conditions, Scripts and Slides

<http://sabin.io/webinar>

Book a slot to let us help you with,

- Integrating SentryOne in your environment
- SQL Server Health Checks, Performance Problems
- Making DevOps work for your Data Platform

<http://sabin.io/bookaslot>

