SQLintersection

Session: Thursday, 1:45-2:45am

Change Your Habits: Tips to Tune Your T-SQL

Grant Fritchey
grant@scarydba.com

Get in touch

Microsoft® MVP Most Valuable Professional

Grant Fritchey

scarydba.com

grant@scarydba.com

@gfritchey

Goals

- **Understand that writing for readability is part of performance**
- **Learn mechanisms for writing T-SQL that performs well**

Test

Test

Test

Test

Test

Remember

- **This is focused on performance tuning**

Remember

- **This is focused on performance tuning**

There are **always** exceptions

Remember

- **This is focused on performance tuning**
- **There are always exceptions**

Your system may be **DIFFERENT**

Remember

- This is focused on performance tuning
- There are always exceptions
- Your system may be different

*Did I mention **test**?*

SQL intersection

Remember

- **This is focused on performance tuning**
- **There are always exceptions**
- **Your system may be different**
- **Did I mention test?**

In
cre
men
tal
changes

Remember

- This is focused on performance tuning
- There are always exceptions
- Your system may be different
- Did I mention test?
- Incremental changes

Monitor your servers

Remember

- This is focused on performance tuning
- There are always exceptions
- Your system may be different
- Did I mention test?
- Incremental changes
- Monitor your servers

You might want to TEST

Remember

- This is focused on performance tuning
- There are always exceptions
- Your system may be different
- Did I mention test?
- Incremental changes
- Monitor your servers
- You might want to test

SQL intersection

Generated T-SQL

Writing for Readability

- **Define a local standard**
  - Object names
  - Comments
  - Format
  - Error handling
- **Enforce the standard**
  - Document it
  - Code reviews

SQL intersection

Object Names & Format

- **Names should be descriptive**
  - Procedures should be a phrase
  - Abbreviations should be common (no Ddltbl)
- **Use aliases**
  - Clear
  - Common
- **Be consistent**
  - A foolish consistency is the hobgoblin of little minds
  - Keyword in that sentence is “foolish”

Comments

- Do something
- Most important is describing the action, not documentation
- Use source control for documentation
- Self-documenting code is a lie

Error Handling

- **TRY/CATCH**
- **Deadlocks**
- **Establish local best practices**

Personal Preferences

- **CamelCase**
- **Uppercase for reserve words and key words**
- **Line breaks**
  - On SELECT list
  - Between JOIN criteria
  - Between WHERE criteria
- **Use the semicolon**
- **Indent**
  - After initial SELECT
  - With ON clause
  - After WHERE clause

Data Types

- **Problem**
  - Implicit or explicit data type conversion
- **Indications**
  - Scans
  - Slow performance
- **Solution**
  - Use appropriate data types

Commands in Comparisons

- **Problem**
  - □ Command on column in WHERE or JOIN criteria
- **Indications**
  - □ Scans
  - □ Slow performance
- **Solution**
  - □ Don’t run commands on columns
  - □ Use sargeable functions

SQL intersection

Improper Use of Functions

- **Problem**
  - Multi-statement user defined functions cause poor performance
- **Indications**
  - Zero cost scan operator in exec plan
  - Very slow performance
- **Solution**
  - When working with more than a few (~50?) rows, don’t use them
  - When using operations such as JOIN that require statistics, don’t use them

SQL intersection

Query Hints

- **Problem**
  - Query hints in the code such as FAST n, index hints, JOIN hints
- **Indications**
  - Inconsistent performance behavior
  - Bad performance
  - Odd looking execution plans
- **Solution**
  - Use of query hints should be exceptional
  - Exceptions are rare, not standard practice

Recompiles

- **Problem**
  - Excessive blocking and CPU contention caused by constant or long statement recompiles
- **Indications**
  - Recompile events in extended events or trace
  - Blocking
  - High CPU usage
- **Solutions**
  - Avoid interleaving DDL & DML
  - Where viable, use table variables

SQL *intersection*

Row by Agonizing Row

- **Problem**
  - Cursors instead of set-based operations
  - WHILE loops instead of set-based operations
  - Multiple statements where 1 will do
- **Indications**
  - Extremely slow performance
- **Solutions**
  - Eliminate unnecessary row-by-row processing

Nested Views

- **Problem**
  - Views within views causes optimizer timeout
- **Indications**
  - Incredibly complex query plans for simple queries
  - Very poor performance
- **Solutions**
  - Don't nest views
  - Materialize views

Relax

- It’s not that bad
- Most things can be changed
- Help is available
- Test everything

Questions?

How would you… ?

What happens when… ?

Why does… ?

When do I… ?

Get in touch

Microsoft Most Valuable Professional

Grant Fritchey

scarydba.com

grant@scarydba.com

@gfritchey

Save the Date

www.SQLintersection.com

24-28 October 2016

We’re back in Vegas baby!

SQL
intersection