|  |  |  |
| --- | --- | --- |
| Slide 1 |  | I’ve been exploring the idea of refactoring a database. To do so, I realized I needed to learn about triggers. I’d thought I’d share what I’ve learned. |
|  | Book reference |  |
| Slide 4 |  | Refactoring a database consists of essentially 3 steps: expand, migrate, and contract.  It fits in well with current programming trends such as Agile and Lean, which emphasize Minimum Viable Product, and experimentation. |
| Slide 5 |  | Expand (optional) |
| Slide 6 |  | Migrate |
| Slide 7 |  | Contract (optional) – the opposite of expand. (Rename instead of drop if you’re really cautious.) |
| Slide 8 |  | Demo:  Create database – show result  Show desired change  Expand (change table)  Convert data  Add trigger. Talk about trigger structure. Inserting, IF, deleting  Show test of trigger  Show effect of trigger when updating from OLD code. Notice extra counts.  (Migrate)  Show effect of trigger when updating from NEW code  Remove trigger, contract  Final results - demonstrate |
| Slide 9 |  | Lessons learned  Discovered the RECURSIVE\_TRIGGER option. By default, It’s off. Learned what happens when it’s on, namely…  Triggers execute even if no rows are changed.  Setting up a test is a lot of work, so have a test script. |
| Slide 10 |  | References |
| Slide 11 |  |  |

**Consider extra slides in case SQL Server fails.**

* Snapshot of old query results BEFORE change. (results and messages)
* Test results from test script for trigger.
* Snapshot of old query results after adding trigger. (results and messages)
* Snapshot of new query results with trigger. (results and messages)
* Snapshot of new query results after trigger. (results and messages)