DRAFT - IT Project Guidance

Development – Database Design Optimisation

Version:

0.1

## Description

Databases schema designs are often the most likely culprit for performance bottlenecks and can often be optimised.

## Synopsis

While system responsiveness is a whole system responsibility and beyond the scope of this guidance paper, Databases can be optimised to contribute to it.

## Contents

[Description 1](#_Toc145232973)

[Synopsis 1](#_Toc145232974)

[Contents 2](#_Toc145232975)

[Synopsis 3](#_Toc145232976)

[Introduction 3](#_Toc145232977)

[Heading Level 3 3](#_Toc145232978)

[Heading Level 4 3](#_Toc145232979)

[Appendices 4](#_Toc145232980)

[Appendix A - Document Information 4](#_Toc145232981)

[Images 4](#_Toc145232982)

[Tables 4](#_Toc145232983)

[References 4](#_Toc145232984)

[Review Distribution 4](#_Toc145232985)

[Audience 4](#_Toc145232986)

[Structure 4](#_Toc145232987)

[Diagrams 4](#_Toc145232988)

[Terms 5](#_Toc145232989)

## Introduction

BOSSCARD/ RAID: Background [], Objective, Options, Scope[In/Out], Stakeholders [Users], Constraints, Assumptions, Risks, Dependencies, Decisions, Deliverables.

At

At a very high level, performance can be summarised with the following pseudo equation:

*(number of catalogues) x (number of indexes) x (number of views) x (size of database) x (users) x (frequency of updates) x (processor) x (network latency)*

Impact

the more indexes you add, the slower your inserts and deletes will go,

But also the more competition pages will have for precious memory space.

Caveat

Even after optimisation.

You may have to throw hardware at it in the form of more memory and faster storage.

Indexing

Missing Indexing[[1]](#footnote-2)

While executing a SELECT statement is faster on a clustered table, INSERTs, UPDATEs, and DELETEs require more time, as not only data is updated, but the indexes are updated also.

if indexes are not properly created, SQL Server has to go through more records in order to retrieve the data requested by a query. Therefore, it uses more hardware resources (processor, memory, disk, and network) and obtaining the data lasts longer.

Indexes improve SELECT, and decrease the speed of INSERT, UPDATE and DELETE operations.

Primary Key

Should be a value that doesn’t get updated or changed afterwards.

Should not be large (blob, varchar(max), etc.).

Should increment so that it can be inserted at the end of the index.

Missing Clustered Index

A table without a clustered index can also be considered as a poor indexing practice.

Incorrect Clustered Index

Using a column value that is naturally ordered (eg: an integer, datetime, datetime based UUID) is correct.

A clustered index accepts nulls, so should be a Primary Key, that does not.

The value of the key should not ever change, as that leads to reordering.

Wasteful index.

Don’t index A, B, C if only filtering on A or AB or AC. If you need to, use secondary indexes for those use cases, while remaining consciounse that the more indexes includes the slower writes.

Clustering on anything else is not

If a column where new values are not higher than previous is used for a clustered index, adding each new row would require re-ordering,

 i.e. moving the whole row and placing it to its proper location in accordance with clustered index ordering, thus splitting data pages and affecting SQL Server performance. If such clustered index is created on a table with frequent inserts and updates, it can cause performance degradation.

Multiple Columns

Avoid indexes created from multiple columns which instead of speeding up queries slow them down.

Wrong Indexing

A wrong index doesn’t provide easier data manipulation or an index created on multiple columns which instead of speeding up queries, slows them down.

Finally, a clustered index should not be built on a column already used in a unique index.

Another thing - when creating a clustered index on a column that has Foreign Key Constraints. Avoid.

Updates hurt more than other operations if they include changes to indexes.

 there are two index modifications (a deletion + an insert) for each data update operation.

**Narrow Indexes** - It’s always better to have small indexes instead of one covering or wide index. Because the data is sorted as per the index columns from left to right, so, if a table has one index including 3 columns in it; let’s say, col1, col2, col3 in same order, any query having joins or where clause on col2, col3 (excluded left most col1), will have much more probability of a table scan instead of using this index. In case of three indexes on col1, col2 and col3, database engine can perform index join and index intersection methods and use these indexes instead of going to scan entire table. Remember, index join may or may not be picked by optimizer. On the contrary Index consolidation is where the criteria for many queries are examined and a number of smaller indexes are intelligently combined into just one or two wider indexes. It's a very effective method for reducing the size of the indexes needed which reduces the database footprint in memory and in backups, etc.

Shape of columns

As a rule of thumb, the columns that are used for decision making in WHERE clauses, and conditions such as greater than (>), less than (<) etc, should be placed before the columns not involved in these clauses. In a case of multiple columns in the WHERE clause, the most distinctive column names should be mentioned earliest in the Index definition.

Shape of Queries

In a case of multiple columns in the WHERE clause, the most distinctive column names should be mentioned earliest in the Index definition.

Foreign Key Constraints

Enforcing foreign key constraints can have a significant impact on database performance, especially on large tables with high write workloads. The additional overhead of checking the constraints can slow down database operations, especially when data is inserted, updated, or deleted.

Do not Referencing data that changes.

Table design

**Query evaluation and columns consideration**- If an index is defined on (col1, col2, col3) and your query has a where clause, such as "where col2 = 'somevalue'",that index won't be used. A non-clustered index can only be used if the firstcolumn in the index is referenced within the where clause.

A where clause, suchas "where col3 = 'someval'", would not use the index, but a whereclause, like "where col1 = 'someval'" or "where col1='somevaland col3 = 'someval2'" would pick up the index. The index would not usecol3 for its seek, since that column is not after col1 in the index definition.If you wanted col3 to have a seek occur in situations such as this, then it isbest if you dehfine two separate non-clustered indexes, one on col1 and theother on col3.

Concluding there are mainly two approaches followed for a good indexing strategy. First will be a **pro-active approach** where we identify the columns(fields involved in JOIN or WHERE) based on the queries and add appropriate indexes. Other one would be the **re-active approach** where we create indexes only when it is required to improve performance after active monitoring.

Appendices

Appendix A - Document Information

### Authors & Collaborators

Sky Sigal, Solution Architect

### Versions

* 1. Initial Draft

### Images

[Figure 1: TODO Image 2](#_Toc144995112)

### Tables

[Table 1: TODO Table 3](#_Toc145048484)

[Table 2: TODO Table 2 3](#_Toc145048485)

### References

* [SQL Server Wait Statistics: Tell me where it hurts (sqlskills.com)](https://www.sqlskills.com/blogs/paul/wait-statistics-or-please-tell-me-where-it-hurts/)
* [SQL Server Performance Tuning Using Wait Statistics | SQLskills](https://www.sqlskills.com/help/sql-server-performance-tuning-using-wait-statistics/)

**There are no sources in the current document.**

### Review Distribution

The document was distributed for review as below:

|  |  |
| --- | --- |
| Identity | Notes |
|  |  |
|  |  |
|  |  |

### Audience

The document is technical in nature, but parts are expected to be read and/or validated by a non-technical audience.

### Structure

Where possible, the document structure is guided by either ISO-\* standards or best practice.

### Diagrams

Diagrams are developed for a wide audience. Unless specifically for a technical audience, where the use of industry standard diagram types (ArchiMate, UML, C4), is appropriate, diagrams are developed as simple “box & line” monochrome diagrams.

### Terms

Refer to the project’s Glossary.

##### IT

: acronym for Information, using Technology to automate and facilitate its management.

##### ICT

: acronym for Information & Communication Technology, the domain of defining Information elements and using technology to automate their communication between entities. IT is a subset of ICT.

1. [Poor database indexing - a SQL query performance killer - recommendations (sqlshack.com)](https://www.sqlshack.com/poor-database-indexing-sql-query-performance-killer-recommendations/) [↑](#footnote-ref-2)