# Relational VS Object Database

May 15, 2017

## 1 Introduction

VelocityDB is a NoSQL object database. Microsoft SQL Server is a relational database.

## 2 Feature Comparison

In table below, support for a feature is one of: ☺ (great) 😐 (ok) ☹ (bad).

| Feature | VelocityDB | SQL Server |
|---|---|---|
| Acid Transactional | ☺ | ☺ |
| Android | ☺ | ☹ |
| Any CPU (32bit/64bit) | ☺ | ☹ |
| Array support | ☺ | ☹ |
| Auto Increment on a field | ☺ | ☺ |
| Backup & Restore | 😐 | ☺ |
| Change event subscription & notification | ☺ | ☺ |
| Choice of data structure to use | ☺ | ☹ |
| Compression of data | ☺ | 😐 |
| Data Fragmentation | ☺ | ☹ |
| Data Integrity options | 😐 | ☺ |
| Database level locking | ☺ | ☺ |
| Distribution ability | ☺ | ☹ |
| Embed ability | ☺ | ☹ |

| Feature | VelocityDB | SQL Server |
|---|---|---|
| Encryption of data | ☺ | ☺ |
| Enum support | ☺ | 😐 |
| High levels of concurrent updates | 😐 | 😐 |
| High Performance | ☺ | ☹ |
| Indexes | ☺ | ☺ |
| In-Memory Only Option | ☺ | 😐 |
| iOS | ☺ | ☹ |
| LINQpad | ☺ | ☺ |
| Linux | ☺ | 😐 |
| No object relational mapping required | ☺ | ☹ |
| Object/Row level locking | 😐 | ☺ |
| Optimistic Concurrency Support | ☺ | 😐 |
| OS X (Mac) | 😐 | ☹ |
| Page level locking | ☺ | ☺ |
| Page level versioning | ☺ | ☹ |
| Pure C#, no other language required | ☺ | ☹ |
| Required Database Administration | ☺ | ☹ |
| Scalability | ☺ | 😐 |
| Small footprint | ☺ | ☹ |
| Store graphs of connected objects | ☺ | 😐 |
| Universal Windows | ☺ | ☹ |

| Feature | VelocityDB | SQL Server |
| --- | --- | --- |
| Variable page size | ☺ | ☹ |

# 3 Pros and Cons

In table below, pros are highlighted yellow and cons are highlighted turquoise

| VelocityDB Pros/Cons | SQL Server Pros/Cons |
|---|---|
| Capable of unbeatable performance and scalability | Simple applications perform well but as data model gets more complex and data size grows performance suffer |
| Use class inheritance, polymorphism and composition | Hard to mimic all object oriented features |
| Limited testing | Very well tested |
| Not very many have used | Many know how to use |
| Use any data structure | Limited to table data structure |
| Standardized Object identifier | Each table defines a primary key |
| No mapping required | An object relational mapping tool such as EntityFramework or Dapper is required |
| Field can store multiple values | Limited to single value in each cell |
| Integrated Client Caching Facility | Client caching has to be done with separate tool(s) |
| No Database Administration Required | Database Maintenance/Administration Required |
| No empty space on pages and fragmentation avoided by using variable page size | Database tables and indexes are usually fragmented with empty/unused space on pages. |
| Good for supporting storage of binary data such as video & audio | Although BLOB storage is supported, data is more difficult to work with and not easy to segment |
| Low Licensing Costs | High Licensing Costs |
| 22,409 lines of C# code | More than 1 million lines of C++/C code |