**The pros of using MongoDB**

MongoDB is a document-oriented database. Data is organized as JSON document’s (rows equivalent) fields (columns equivalent) which are assembled into collections (tables equivalent).

MongoDB has the absence of referential integrity (RI) but this does not mean that it is not a relational database. It has a built-in capability to enforce relations between different pieces of data in the database (e.g. foreign key constraints). However, we need to consider that Relational databases, provide strict data integrity enforcement and reliable way of combining the records during fetching.

Since the release of version 4.0, we finally have multi-document transaction support (ACID) mirroring the one we have known from relational counterparts.

In MongoDB, there is no direct way to join additional documents into another collection during fetching. However, there are separate ways to aggregate data by $lookup (combining documents from multiple collections) and aggregation pipeline (grouping, filtering and processing of documents from one collection).

It also recommended that sometimes embed selected documents instead of creating separate collections for them. But embedding too much information in one document will result in slower query processing. It is imperative to make sure that there is an ORM library (Object-Relational Mapping) for our programming language. The most popular are: mongoose (for node.js) and mongoid (for Ruby on Rails).

The document-oriented database has positive advantages. Its flexibility (lack of rigid structure), (direct use of JSON), big data processing and real time statistics/data analysis.