

Relational database

Since the 1960 it has been well known that SQL databases are the way to go for complex systems storage.

TS is based on the MySQL / MariaDB / Percona databases, and all data is fully normalized granting multiple benefits

- High performance with high volumes
- Easy to integrate with BI systems
- Predictable structures

Scalable applications

Applications built in TS will adhere to common design practices ensuring good performance, stability and integrity.

The platform is very scalable in regard to

Complexity

You can keep extending the models as far as needed, using the Parent and Child relations.

Due to a very elaborate and flexible permission model, multiple user types and scenarios can easily be supported on the same setup.

We even support logical constructs such as many-to-many relations. TS has customers running more than 350 linked applications on the same server.

Performance

TS databases are largely unaffected by data set sizes and number of concurrent users.

There building configurations for indexing fields, and the built in searches can be supported by Elastic search and indexing.

To prove our point we have actually had the whole platform running smoothly on a Raspberry Pi.

Volumes

Databases will handle large amounts gracefully, but large object sizes will often cause problems in back/restore scenarios.

TS supports unlimited file size and count, using a mixed database and block storage approach.

Relational searches

A common problem querying data is specifying criterions and values in different parts of the model

Show orders where the sum > 10000, and one or more ordelines refers to the product Smart TV

In TS this is handled gracefully by allowing interlinked searches

Look at parents referring children

1. Make a search in the child items (optionally name the view)
 - Set search criterions normally
2. Make another search in parent item
 - Set search criterions normally
 - In the *List of children* field refer to the CURRENT_QUERY (or given name)

Look at children referring parents

1. Make a search in the parent items (optionally name the view)
 - Set search criterions normally
2. Make another search in child items
 - Set search criterions normally
 - In the *Parent reference* field refer to the CURRENT_QUERY (or given name)

Revision #2

Created 8 April 2025 12:44:06 by Theis Villumsen

Updated 13 October 2025 12:11:01 by Max Gøtske