.ai
How to select the optimal database in predictive analytics projects
Tyris AI Team

How to select the optimal database in predictive analytics projects

tyris software (https://tyris-software.com/en/author/vlcteche/) - April 28, 2020

.ai (https://tyris-software.com/en/industry)

Authorship
Tyris AI Team

Share:

(https://www.facebook.com/sharer.php?u=%20http://tyris-software.com/how-to-select-the-optimal-database-in-predictive-analytics-projects)

(https://twitter.com/share?text=https://tyris-software.com/how-to-select-the-optimal-database-in-predictive-analytics-projects)

(https://www.linkedin.com/shareArticle?url=https://tyris-software.com/how-to-select-the-optimal-database-in-predictive-analytics-projects)

Here at Tyris.AI (http://tyris.ai) we develop predictive analytics systems for industry. We capture data at high frequencies, sometimes from several hundreds of sensors at once, and store them securely and rapidly (in real time). This means we need r**obust methods for large-scale data input.** This is a challenge by itself, but it is further composed by the fact that the sensors we track might change during the development and lifecycle of the system. Thus, the **database structure for our solution must change in kind**.

It is quite understandable that we pose the question: what database typology will best solve all these problems?

Before we answer the question, let us keep in mind that, aside from data input, **the chosen databases for predictive**

analytics must allow for efficient access in relation with the process. Intelligent models are nurtured by and learn from data. Thus, **generating a sufficient and adequate *dataset* to train or re-train these algorithms** can mean working with millions of data, creating a need for agile systems.

Tyris.AI (http://tyris.ai) is quite conscientious when choosing the particular database to use for each project, depending on the productive scenario where they are going to be implemented.

Sometimes, it is quite difficult to choose the optimal database structure that can allow for both rapid data input and rapid data querying. Thus, for predictive analytics solutions, it is important to **analyze multiple database typologies and compare their advantages and disadvantages.**

Let us analyze some of these typologies:

TYPOLOGIES

Relational:(https://developer.android.com/google/play/expansion-files)

These are the ones that come to mind when speaking of databases, since they are the most widely used. The best-known examples are **SQL databases, such as SQLServer, Oracle, and PostgreSQL. They can model complex relations among data and tables** by means of primary and foreign keys.

Generally speaking, they have powerful data input and querying systems that exploit the complexity of the data and their relations. **They can also verify the integrity of the data and avoid duplicities.** Relational databases, however, are known to have low flexibility and scalability.

Non-Relationals:(https://developer.android.com/google/play/expansion-files)

They are the alternative to relational databases that have attracted the most attention in the past few years. Notable examples are **MongoDB, Cassandra, and HBase. They can work with different data structures more flexibly than relational databases.** However, this flexibility usually comes at the expense of a reduced capacity to model and navigate complex relations in the data. Furthermore, the data are less structured and present worse integrity.

Since there is **no need to maintain a pre-defined data structure, non-relational databases almost always have rapid input and querying speeds and greater capacity** to work with large volumes of heterogeneous data. Because of this, they are especially suitable for **projects or productive environments where there are different data types to integrate** (e.g., other than numeric) and the number of variables to integrate can change.

Time series databases (https://developer.android.com/google/play/expansion-files)

This typology is probably the least known, since it is **highly specialized.** As their name suggests, time series databases are optimized to **manage time series data** (data streams). Some of the best-known examples are

InfluxDB, TSI by Microsoft Azure, or Amazon Timestream. In industry, and more generally in IoT environments, most of the available information to integrate comes from sensors that produce **data as time series.** The data integrated in these databases are time indexed with the required precision. This way, the can be correlated across different input dimensions (corresponding to the different monitored sensors). The data can be stored once or periodically

One of the main features of time series databases is that **their input speed is stable and much larger than their query speed.** Furthermore, they tend to provide limited data update capabilities.

Real-time databases (https://developer.android.com/google/play/expansion-files)

Lastly, **real-time processing** is another important requirement in some industry applications. Real-time databases are adapted to work in **environments with high temporal demands, where data must be stored and analyzed as they come, usually within milliseconds.** These demands place particular emphasis on the window of time where data are valid, in order to avoid making decisions using data that was already "stale" at the time of analysis. These speeds are made possible **by storing data in memory, rather than disk.** Persistence can be achieved by storing the data in disk after the real-time processing has concluded. One of the best-known real-time databases is Redis. **Redis** is one of the youngest databases out there but it is gaining traction as the number of implementations rise.

Thus, once we know the specifications of each typology, we are ready to choose the best database structure for a given project. Another important aspect to consider is the wide array of brands and standards available in the market. **They can include extra tools to help us with installation, configuration, and inter-system interoperability.**

Finally, we will touch upon some databases that are suitable for industry environments, **according to our own experience and the features mentioned above:**

DATABASES SUITABLE FOR INDUSTRY

MongoDB (https://developer.android.com/google/play/expansion-files)

It is an ideal option for changing environments, e.g., development projects where the number of tracked signals or the frequency at which they are sampled can change with time. **MongoDB allows to store signals or data sequences directly.** However, its query system is less flexible than that of other options.

Cassandra (https://developer.android.com/google/play/expansion-files)

Cassandra allows for **fast queries and facilitates horizontal scalability** to include new variables or manage distributed databases. In our opinion, it is the ideal option for **mature systems** where **the data can be stored in the same (or similar) format that we will need during data consumption for processing,** thanks to its column-based and key-value configuration.

Redis (https://developer.android.com/google/play/expansion-files)

The best option for real-time, memory-based work. Although this mode is not persistent, the data can be stored for later processing or they can be kept in memory for immediate use, e.g., in ***dashboards that monitor and control a system.***

PostgreSQL (https://developer.android.com/google/play/expansion-files)

Its **powerful query system is optimal for environments with stable and homogeneous data structures,** and almost every developer is familiar with SQL. However, it is not very flexible when it comes to structure changes. This makes it interesting to store already-processed data, after their usage in predictive analytics systems, and relation these data with the rest of values in the system environment.

To summarize, **we believe it is quite hard to choose a single database typology that can produce efficient results in all scenarios.** Thus, in order to choose the optimal typology, it is quite important to carefully analyze the environment and the current and future requirements of the system. With the above considerations, we should be at least a bit more ready to tackle the challenge. **Are you ready?**

(https://tyris-software.com/en/tv/)

Autoría
Equipo Tyris AI

Comparte esta página:

(https://www.facebook.com/sharer.php?u=%20http://tyris-software.com/como-elegir-la-base-de-datos-optima-en-proyectos-deanalitica-de-predictiva)

(https://twitter.com/share?text=https://tyris-software.com/como-elegir-la-base-de-datos-optima-en-proyectos-deanalitica-de-predictiva)

(https://www.linkedin.com/shareArticle?url=https://tyris-software.com/como-elegir-la-base-de-datos-optima-en-proyectos-deanalitica-de-predictiva)

TAGS: BIG DATA (HTTPS://TYRIS-SOFTWARE.COM/TAG/BIG-DATA/), DATABASE (HTTPS://TYRIS-SOFTWARE.COM/TAG/DATABASE/), DATASET (HTTPS://TYRIS-SOFTWARE.COM/TAG/DATASET/), INDUSTRIA (HTTPS://TYRIS-SOFTWARE.COM/TAG/INDUSTRIA/)

Previous Post
Design of databases in high concurrency backends (https://tyris-software.com/en/databases-in-high-concurrency-backends/)

(mailto:info@tyris-software.
Paseo Facultades 1 B, Prime
email: info@tyris-software.c
phone: +34 961 152 302

(https://sede.micinn.gob.es/pyiINFO/buscarPyi.mec?&nif=B98628282)