1. Reddis Database

Reddis is a noSql database and it is an in-memory dataset can be used to store data such as strings and list.

Reddis works as: Key -> [Value, …]

For example, you can have a product name and ID. Eg: Panado -> [83] (key->[value]).

At the first on my testing I wanted to have the key as the ID of the product on the Firebird DB (Main DB) and value as the productName, but it gives a problem because Reddis db only performs search on the key and not the value.

So I am now storing inside the key the productName then value the ID of the product, because all product names have a unique ID number.

So Reddis when you add an element with the same key it just adds on the value, it doesn't repeat values.

For example:

I insert: Panado->83 then I insert again panado->100, 83 for Panado Tab and 100 for panado Syrup (just an example) So inside the database it will look like this:

Panado->[83,100].

So because the key can take regular expression to improve the searching it actually helps a lot.

I like this db because you can easily add on it. Let say I want to search for panado but I want to call it with my own name let say Pa-nado I can give it the value 83 as an ID, so when I later search for it, it will bring back 83 which is the same with "Panado". And I can add as much as I want. SO it is very scalable.

Unfortunately, it is not a relational DB, hence using this for a faster and more scalable data structure means we must use both Firebird and this one. In the sense of all search and search logs are done or added into reddis. But the ID(s) that it brings back are used to fetch the product information on Firebird database (product schedule, dosage form, description, etc.)