4 Benefits of Integrating a Language With a Database

By Dr. Jans Aasman:

One of the hallmarks of a truly modern programming language is the coupling of a language—for creating applications, devising algorithms and solving business problems—with an underlying database. When there is practically no separation between a database and its programming language, developers immensely increase their productivity, maximize the effectiveness of the code they write and provide unparalleled speed and flexibility in supporting business needs.



Predicating a programming language on a database delivers these advantages by enabling developers to treat data as if it is in-memory, liberating them from the burden of manipulating how data is represented in the underlying database. This one simple advantage exponentially increases the flexibility for addressing business problems and, when used with advanced logic languages, enables developers to write applications in a fraction of the time and with less effort than it would otherwise require.

Specific benefits of tightly coupling a database with its programming language include eliminating the time-honored mismatch between data structures and database representations, smart caching with transactional technologies for committing or rolling back database changes and automatically changing object definitions.

These benefits remove the database as an otherwise complicating factor in supporting business use of data by

freeing developers to flexibly innovate solutions—and speedily implement them.

In-Memory Manipulations

Developers focus on manipulating data structures to solve business problems. Traditionally, this concern was hamstrung by the onus of translating those structures into how data are represented in the database, a time-consuming necessity detracting from programmers' attention to business problems. By fully integrating an object-oriented database or graph database with its programming environment, users can manipulate data objects in memory and they'll automatically persist to storage. Developers can solely focus on working with data structures for business objectives without spending just as much, if not more, effort wrangling data's representation in the database.

Read the full article at DevOps.