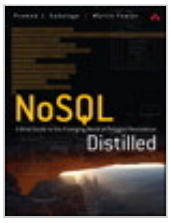
Introduction to Polyglot Persistence: Using Different Data Storage Technologies for Varying Data Storage Needs

By Pramod J. Sadalage and Martin Fowler

Sep 5, 2012

Different databases are designed to solve different problems. Using a single database engine for all of the requirements usually leads to non- performant solutions. In this chapter, the authors introduce the term polyglot persistence to describe using different data storage technologies to handle varying data storage needs.

This chapter is from the book



NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence

Learn More Buy

Different databases are designed to solve different problems. Using a single database engine for all of the requirements usually leads to non-performant solutions; storing transactional data, caching session

information, traversing graph of customers and the products their friends bought are essentially different problems. Even in the RDBMS space, the requirements of an OLAP and OLTP system are very different—nonetheless, they are often forced into the same schema.

Let's think of data relationships. RDBMS solutions are good at enforcing that relationships exist. If we want to discover relationships, or have to find data from different tables that belong to the same object, then the use of RDBMS starts being difficult.

Database engines are designed to perform certain operations on certain data structures and data amounts very well—such as operating on sets of data or a store and retrieving keys and their values really fast, or storing rich documents or complex graphs of information.

13.1. Disparate Data Storage Needs

Many enterprises tend to use the same database engine to store business transactions, session management data, and for other storage needs such as reporting, BI, data warehousing, or logging information (Figure 13.1).

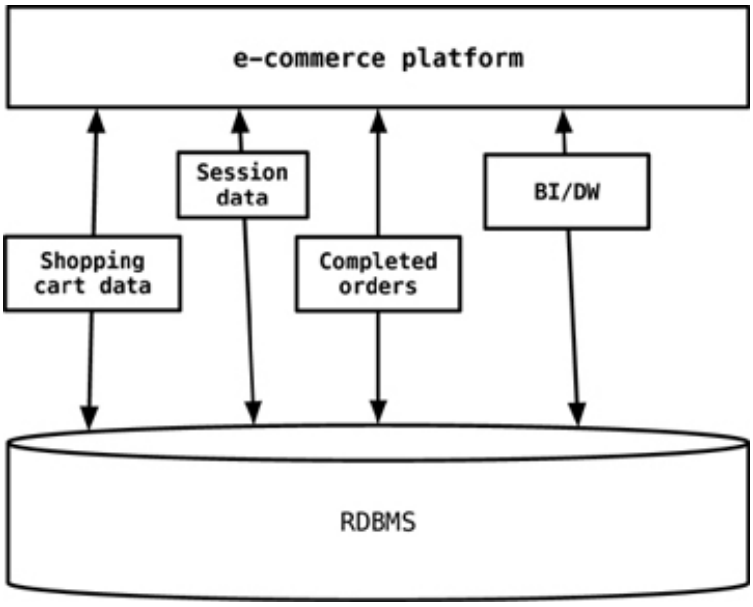


Figure 13.1 *Use of RDBMS for every aspect of storage for the application*

The session, shopping cart, or order data do not need the same properties of availability, consistency, or backup requirements. Does session management storage need the same rigorous backup/recovery strategy as the e-commerce orders data? Does the session management storage need more availability of an instance of database engine to write/read session data?

In 2006, Neal Ford coined the term **polyglot programming**, to express the idea that applications should be written in a mix of languages to take advantage of the fact that different languages are suitable for tackling different problems. Complex applications combine different types of problems, so picking the right language for each job may be more productive than trying to fit all aspects into a single language.

Similarly, when working on an e-commerce business problem, using a data store for the shopping cart which is highly available and can scale is important, but the same data store cannot help you find products bought by the customers' friends—which is a totally different question. We use the term **polyglot persistence** to define this hybrid approach to persistence.

+Share This Save To Your Account **Page 1** of 8 Next >

Comments for this thread are now closed.

Comments Community 1 Login

Recommend Sort by Oldest

This discussion has been closed.