
Article

[Developer Commu...](#) · Oct 21, 2015 1m read

Fighting Impedance Mismatch at the Database Level

Introduction

With the maturation and wide acceptance of Java, object-oriented programming has moved to the foreground of the application development landscape. Because of their rich data models and support for productivity-enhancing concepts such as encapsulation, inheritance, and polymorphism, object technologies like Java, C++, and COM, are favored by today's application developers.

However, much of the world's data still resides in relational databases. Developers of database applications (that is, any application that accesses stored data) often find themselves fighting impedance mismatch: the inherent disconnect between the object and relational data models. Efforts to "map" relational data into a usable object format are often detrimental to both programmer productivity and application performance.

However, impedance mismatch can be mitigated by the proper choice of database technology. This paper defines impedance mismatch and gives two simple examples of how it affects application development. It then discusses, with regards to impedance mismatch, the pros and cons of three kinds of database: relational, object, and Caché, the multidimensional database from InterSystems.

[#Caché](#) [#SQL](#)

Source URL: <https://community.intersystems.com/post/fighting-impedance-mismatch-database-level>