The Data Access Object (DAO) pattern is a mechanism to abstract away the details of persistence in an application.

The idea is that instead of having the domain logic communicate directly with the database, file system, web service, or whatever persistence mechanism your application uses, the domain logic speaks to a DAO layer instead.

This DAO layer then communicates with the underlying persistence system or service.



In short the purpose of DAO's is to hide the persistence mechanism of an application from its domain logic.

Advantage

If you need to change the underlying persistence mechanism you only have to change the DAO layer. i.e. you can change the persistence mechanism without affecting the domain logic. In fact you might be able to cleanly swap in a new DAO layer for your new database or alternate persistence mechanism.

* The DAO layer usually consists of a smaller set of classes, than the number of domain logic classes that uses it.
* It is also a somewhat more controlled operation, since you can search for all DAO classes, and make sure they are changed to use the new persistence mechanism.

For this encapsulation of the underlying persistence mechanism to work it is important that no details of the underlying persistence mechanism leak out of the DAO layer. Ensuring this is, however, a bit of a challenge.

DAO Design Problems

In practice however, it itsn't always that easy to make your DAO's fully hide the underlying persistence layer.

For instance, when a transaction span calls to multiple DAO's, where do you put the transaction demarcation code?

DAO Connection Scoping

method scope - should each DAO method open and close its own connection.

Issue1 - when one DAO method needs to call another.

In that case 2 connections are opened: One in the calling method, and one in the method called. This is a waste of connections.

Issue2 - if the two methods are to run in the same transaction. Then the two methods must use the same connection.

instance scope - should the DAO instance have a single internal connection that all methods use? Or

thread session scope - should connections be shared between all DAO's used by the same thread for the same session?