

The ability to retrieve persistent data from a database is one of the powerful and useful aspects of DevForce. At the most basic, such retrieval occurs in one of two ways, either via a **query** or via a [navigation](#) from one entity to another or to a collection of entities. In both of these cases, the underlying mechanism is the same; a query is submitted to and executed by an [EntityManager](#).

While the most common example of a query is one that operates against a backend relational database, queries may also be constructed so that they operate against a set of data stored in memory on a local machine. One of the most common kinds of query is one that operates against both a backend database as well as a local set of entities with the results combined. This form of query allows a developer to create views of data that represent work in progress.

Queries come in many flavors and may be customized in a variety of ways. The one unifying feature of every DevForce query is that they are all instances of some subclass that implements the *IEntityQuery* interface. The *EntityManager* has a variety of methods that operate on any query type that implements this interface.

The remainder of this chapter will enumerate:

- [LINQ](#), [non-LINQ](#), and [asynchronous query](#) composition.
- [How to control where and how a query is executed](#).
- [How to control the manner in which the results of a query are merged with data already on the client](#).
- [The mechanisms by which entity navigation queries may be customized and mediated](#).
- [The lifecycle of a query and the points at which query execution can be intercepted](#).
- [Ways to improve the performance of your queries](#).