Querying using property navigation is a convenient syntax for accessing data from *related* entities.

Consider these familiar scenarios:

- Get the line items (*OrderDetail*s) of an order
- Get the name of the order's parent customer
- Get the parent customer's other orders

In DevForce you might write:

- *anOrder.OrderDetails*
- *anOrder.Customer.CompanyName*
- *anOrder.Customer.Orders*

In each case, we want information (*OrderDetail*s, *Customer*s) related to a single entity (*Order*). The desired information exists somewhere in the entity’s object graph – the network of other entities that are related to it. Property navigation helps us traverse the network to reach related entities.

The third example shows how you can chain navigation properties together. The statement *anOrder.Customer.Orders* navigates from a particular *Order* up to its parent *Customer* and then down to all of the parent *Customer*'s other *Order*s. We call this "walking the object graph".

There are two kinds of navigation property:

1. **Scalar Navigation** returns at most a single entity, typically a "parent" entity as in *anOrder.Customer*.
2. **Collection Navigation** returns a list of related entities, typically the "children" of this entity as in *aCustomer.Orders* and *anOrder.OrderDetails*.