ForeignKeyRepresentationAndManagement

Database table foreign keys map to DataObjects as relationships to other DataObjects. Consider two tables: DEPARTMENT and EMPLOYEE and assume the EMPLOYEE table has a FK column DEPT_ID that references a row in the DEPARTEMENT table.

If you us the DAS to read a particular department and all of its related employees then you will end up with a DEPARTMENT DataObject that has a property named "EMPLOYEE" that is a reference to a list of related EMPLOYEE DataObjects. You can navigate dorectly to the employees via this property like this: department.getList("EMPLOYEE").

Although the relationship is mapped to the DataObjects as a relationship, the FK value is also stored in the repsective DataObject. So, continuing with this example, the EMPLOYEE DataObject will have a simple property named "DEPT_ID" that holds the value of the FK. This property is not maintained after the read. So, if an EMPLOYEE DataObject is moved to a different department, the FK property in EMPLOYEE will not be modified. Although most applications will ignore this FK property, there are a few scenarios that make use of it. In fact, although the preferred method of modifying relationships between DataObjects in a data graph is to use normal POJO manipulation, the DAS does allow the modification of a relationship via direct modification of the FK value.

When the DAS is scanning changes made to a data graph, it employs the following set of rule to guide the generation of update statements:

"The value for a foreign key column will come from either the primary key of the parent dataobject or from the FK property itself. If the relationship is set but the FK property is not, the value will be taken from the parent's primary key. If the FK property is set but the relationship is not, the value will be taken from the property value. If both the relationship and the FK property are set, an exception will be thrown stating that "Foreign key properties should not be set when the corresponding relationship has changed"