Is Dataset an ORM?

Asked 15 years, 9 months ago Modified 12 years, 10 months ago

Viewed 3k times

▲

8

▼

I am a little bit confused about Dataset compared to ORM (NHibernate or Spring.Net). From my understanding the ORM sits between the application layer and the database layer. It will generate the SQL commands for the application layer. Is this the same as what Dataset does? What is the difference between the Dataset and ORM? What are the advantages and disadvantages for these two methods? Hope the experts in here can explain something.

Thanks, Fakhrul

orm dataset

Share

Improve this question

Follow

edited Mar 18, 2009 at 9:01

Ian Nelson

58.6k ● 20 ● 78 ● 104

asked Mar 18, 2009 at 8:12

Fakhrul

121 ● 2 ● 5

6 Answers

Sorted by: Highest score (default)

12

There is a BIG difference between them, first of all about the programming model they represent:

1. The **Dataset** is based on a Table Model
2. An **ORM** (without specify a particular product of framework) is based and tends to a Domain Model.
3. There is another kind of tool which could be used in data scenario, this kind of tool is a **Data Mapper** (eg. iBatis.NET)

As others answers before me, I think it's important to view what Microsoft says about Dataset and better what Wikipedia says about ORM, but I think (this was for me at beginning) it's more to understand the difference between them in terms of model. Understanding that will not only clarify the choises behind but better, will do too easy to approach and understand a tool itself.

As little explanation it's possible to say:

Table Model

is a model which tends to represent tabular data in a memory structure as close as possible (and even as needed). So it's easy to find implementations which implements concepts as **Table**, **Columns**, **Relations** in fact the model is concetrate on the table structure, so object orientation is based on that not on data itself. This model could has its own advantages, but in some case could be heavy to manage and difficult to apply concepts on contained data. As previous answers says, implementations like Dataset, let, or better, force you to prepare (even if with a tool) needed SQL instructions to perform actions over the data.

ORM

is a model which (as **mendelt** says before me..) where **Objects** are mapped directly to database objects, principally Tables and Views (even if it's possible to map even functions and procedures too). This is done in 2 ways generally, with a mapping file which describes the mapping, or with (in case of .NET or Java) code Attributes. This model is based on **Objects** which represents the data, so object orientation could be done on them as in normal programs, it's clear with more attention and caution in certain cases, but generally, when you are confident with ORM it could be a really powerfull tool! Even ORM could be heavy to manage if it's not managed and designed well, or better understood weel, so it's important to understand techniques, but I can say with my experience that ORM is a really powerfull tool. In ORM, the tool principally it's responsible to generate the SQL instructions needed as operations are done in code, and in more cases ORMs has a middle language (like HQL) to perform operations on Objects.

MAPPER

A mapper is a tool which doesn't makes things like an ORM, but, maps hand written SQL instructions to an Object Model. Thi kind of tool could be a better solution when it's needed to write by hand SQL instructions but It's wanted to designe an application Object model to represent data. In this "model" objects are mapped to instruction and described in a mapping file (generally an Xml file as iBatis.Net or iBATIS (java) does). A mapper let you define granular rules in SQL instructions. In this scenario could be easy to find some ORM concepts as for example session management.

ORM and Mappers let to apply some very interesting Design Patterns, which could be not so easy to apply in the same way to a Table Model and in this case to a Dataset.

First of all excuse me for this long answer and about my poor english, but for me, an answer like this makes me in past to understand well the difference between this models and then between implementations.

Share Improve this answer

Follow

edited Feb 22, 2012 at 13:31

kfmfe04

15.3k ● 15 ● 81 ● 144

answered Mar 18, 2009 at 10:09

Hoghweed

1,948 ● 1 ● 16 ● 36

4

the Dataset class is definitly not an ORM; an ORM maps relational data with an object oriented representation.

It can be regarded as some kind of 'unit of work' though, since it keeps track of the rows that have to be deleted/updated/inserted.

Share Improve this answer Follow

answered Mar 18, 2009 at 9:05

Frederik Gheysels

56.9k ● 11 ● 102 ● 155

1

- **ADO.NET DataSet** =

  http://msdn.microsoft.com/en-us/library/zb0sdh0b(VS.80).aspx

- **ORM** =

  http://en.wikipedia.org/wiki/Object-relational_mapping (Example Developer Express XPO,DataObjects.NET)

Share Improve this answer Follow

answered Mar 18, 2009 at 8:24

abmv

7,108 ● 17 ● 64 ● 105

1

ORM is based on mapping between objects and tables. Not the case for this dataset. Dataset is itself in a way directly to the table. ORM is based on a minimum of SQL script. But enough to use the dataset you write SQL clause. Dataset in this case is not an ORM.

Look at dataset and ORM.

Share Improve this answer Follow

answered Mar 18, 2009 at 8:27

rayyildiz
338 ● 3 ● 7

1

No, Datasets are not ORM's. They may look like orms because datasets map tables to objects just like ORM's the main difference lies in what objects they map to.

Datasets have their own table and row object types that closely resemble the structure of the database. You're rebuilding part of the database's relational model in objects. Restricting these objects into something resembling a relational database gets around some of the problems inherent in mapping a database to an object model.

An ORM maps the tables and rows from the database into your own object model. The structure of your object model can be optimized for your application instead of resembling a relational database. The ORM takes care of the difficulties in transforming a relational model into an object model.

Share Improve this answer

Follow

edited Mar 18, 2009 at 10:07

answered Mar 18, 2009 at 9:27

Mendelt

37.5k ● 6 ● 75 ● 97

▲

0

▼

DataSet is a DTO, a data transfer object. DataSet itself can't do anything. You can use a DataAdapter (of the provider used) to produce sql or call predefined queries, though it still isn't doing anything.

Share Improve this answer

Follow

answered Mar 18, 2009 at 10:12

Frans Bouma

8,327 ● 1 ● 29 ● 29