

Object Relational Mapping in Delphi

Malcolm Groves

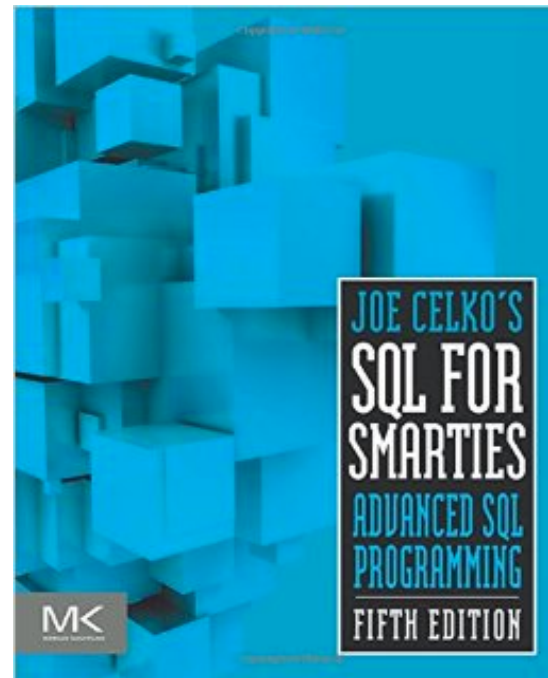
malcolm@code-partners.com

All slides and source available at <https://www.code-partners.com/adug2016/>

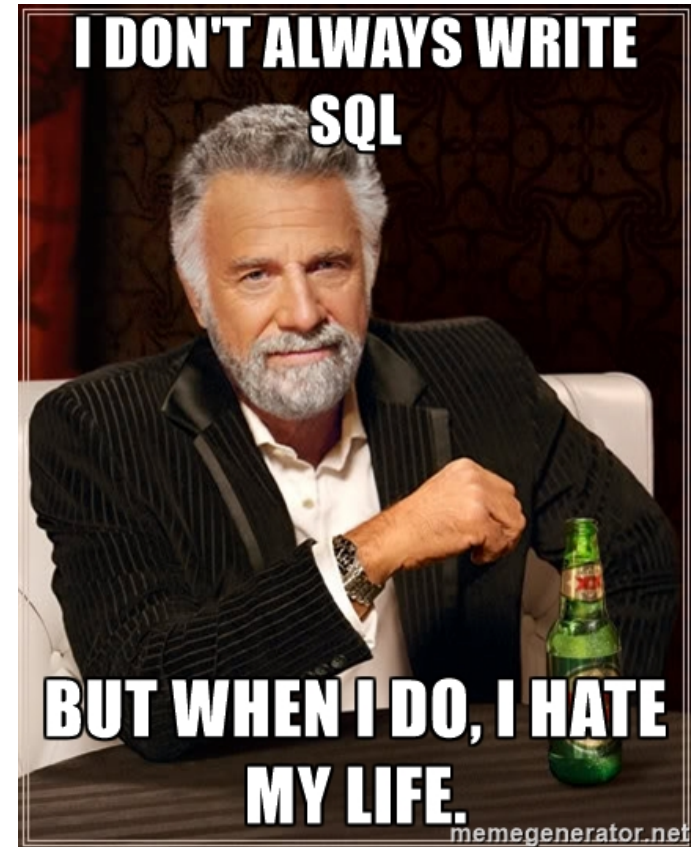


The last time I found SQL interesting...

- 1997, first project for Borland.
- Implementing hierarchical structures in the DB, using this book.
- Nearly 20 years since then of boring, no value-add effort.



- Most SQL you write could be auto-generated on demand.
- If you don't write it, you don't have to maintain it.
- We only write low-level API code when we really have to, SQL should be the same



Framework Options for Delphi

- TMS Aurelius <http://www.tmssoftware.com/site/aurelius.asp>
- EntityDAC <https://www.debart.com/entitydac/?gclid=CLvvir-Fpc4CFYKWvAodpUUNSg>
- tiOPF <http://tiopf.sourceforge.net/index.shtml>
- DORM <https://github.com/danieleteti/delphi-orm>
- iORM <https://github.com/mauriziodm/iORM>
- Write your own
 - Just kidding. Please don't do this.

Common Objections

- “They are too complicated”
- “Performance sucks”
- “We need more control over how our objects are stored”
- “They’re fine for a new app, but we have an existing DB structure so can’t use them”
- “Takes too much time. We need to move faster than that”

Why use an ORM?

Speed!!!

Quick to develop

Quick to debug

Most of the time, quick at runtime.

What I look for in an ORM

- Low ceremony, Gets out of the way
- Flexible Query Language
- Allows control over how it saves.
- Allows control over how\when it loads.
- Works with non-ORM code
- Works with existing database structure

Aurelius – Create your objects

- Plain old Delphi objects
 - Don't have to descend from any particular class
 - Don't have to implement any particular interface
- At a minimum, add two attributes:
 - [Entity]
 - [Automapping]
- You're done.
- Demo

Aurelius – But I want more control

- AutoMapping is letting Aurelius control DB structure
- Can selectively control:
 - Table Name

```
[Table('Customers')]  
TCustomer = class(TObject)  
private
```

Aurelius – But I want more control

- AutoMapping is letting Aurelius control DB structure
- Can selectively control:
 - Table Name
 - ID Datatype (even composite ID's)

```
[Id('FId', TIdGenerator.IdentityOrSequence)]  
TCustomer = class(TObject)  
private  
    [Column('CUSTOMER_ID')]  
    FId: integer;
```

Aurelius – But I want more control

- AutoMapping is letting Aurelius control DB structure
- Can selectively control:
 - Table Name
 - ID Datatype (even composite ID's)
 - Column
 - Eg. Name, Required, NoInsert, NoUpdate, Lazy, Length, Precision

```
[Column('MEDIA_NAME', [TColumnProp.Required], 100)]  
property MediaName: string read FMediaName write FMediaName;
```

Aurelius – But I want more control

- AutoMapping is letting Aurelius control DB structure
- Can selectively control:
 - Table Name
 - ID Datatype (even composite ID's)
 - Column
 - Eg. Name, Required, NoInsert, NoUpdate, Lazy, Length, Precision
 - Association
 - Eg. Lazy, Required, Cascade, JoinColumn, OrderBy

```
[Association([], CascadeTypeAllButRemove)]  
[JoinColumn('ID_SONG_FORMAT', [])]  
property SongFormat: TSongFormat read FSongFormat write FSongFormat;
```



Aurelius – But I want more control

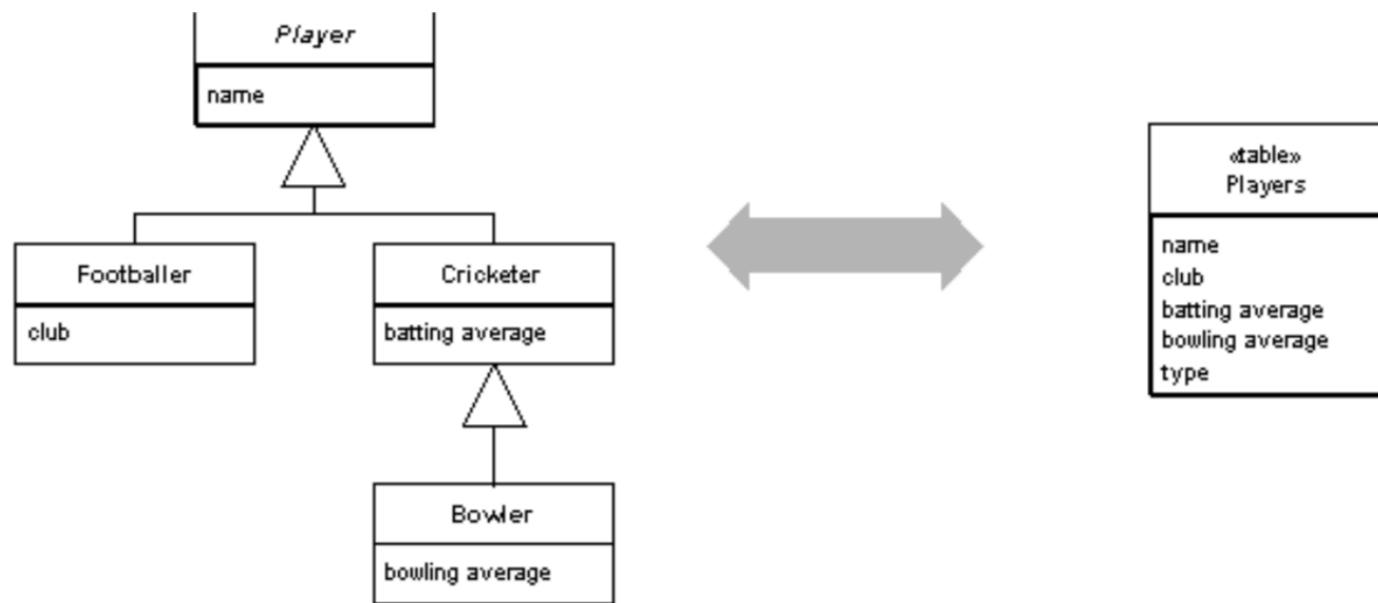
- AutoMapper is letting Aurelius control DB structure
- Can selectively control:
 - Table Name
 - ID Datatype (even composite ID's)
 - Column
 - Eg. Name, Required, NoInsert, NoUpdate, Lazy, Length, Precision
 - Association
 - Eg. Lazy, Required, Cascade, JoinColumn, OrderBy
 - Inheritance Strategy
 - SingleTable, JoinedTable

```
[Inheritance(TInheritanceStrategy.SingleTable)]  
[DiscriminatorColumn('MEDIA_TYPE', TDiscriminatorType.dtString)]  
TMediaFile = class
```



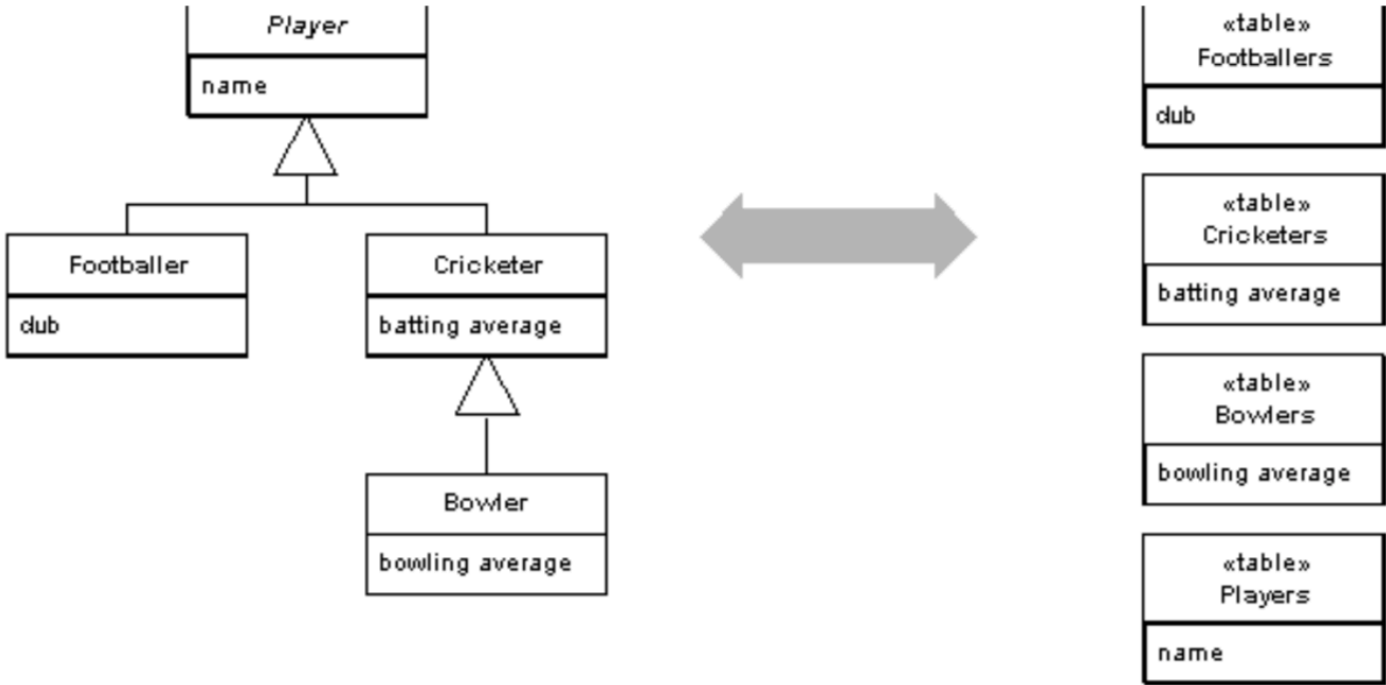
Aurelius – Inheritance Strategy

- Single Table



Aurelius – Inheritance Strategy

- Joined Tables



Aurelius – Create\Update your DB structure

- DatabaseManager.BuildDatabase\UpdateDatabase
- Demo
- Can just call ValidateDatabase to check structure and get a list of issues
- Can also get it to generate a SQL script that will do the migration

Aurelius – Save some objects

- Demo
- `TObjectManager.Save\Update` just update them in `ObjectManager`'s cache
- `TObjectManager.Flush` actually pushes out to the DB

Aurelius – OK, let's get them back

- Demo
- Queries can be polymorphic
- Can also have complex queries:
 - Logical combination of criteria
 - Sub-Criteria
 - Projections
- If you really feel you need to write SQL, go for a walk. You probably don't.
- But if you really, really do need to, you can.

Aurelius – Control Loading

- Fetch using a Cursor
 - On demand creation of Objects
 - Keeps DB Connection open
- Lazy Loading
 - Object is not retrieved until referenced.
 - For Blobs
 - For Relationships

Try it. It's liberating!



Where to next?

- Aurelius Trial version : <http://www.tmssoftware.com/site/aurelius.asp>
- My videos/articles: <https://www.code-partners.com/tag/aurelius/>
- The Aurelius Manual is actually pretty good:
http://www.tmssoftware.biz/business/aurelius/doc/aurelius_manual.pdf
- Aurelius Forum :
http://www.tmssoftware.com/site/forum/forum_topics.asp?FID=36&title=aurelius
- Slides and Source <https://www.code-partners.com/adug2016/>

- If you do decide to buy, Code Partners are doing 10% off
<https://www.code-partners.com/special-offers/>

