

ORM & SQL Primer

Cameron Goodale

@sigep311

Feb. 17, 2015

slides: <http://bit.ly/iepSQLA>

Overview

What is SQL?

What is an ORM?

Design Patterns

Final Warning

What is SQL?

Structured Query Language

- ✓ Talk to a Relational Database
- ✓ **Create, Read, Update, Delete** (CRUD)
- ✓ DB Vendor syntax can differ

Typical Relational Databases

- ✓ MySQL / Oracle (Oracle)
- ✓ Postgres
- ✓ SQLite3
- ✓ SqlServer (Microsoft)

What is an ORM?

Object-relational **M**apping

- ✓ Use OO constructs to work with a RDBMS

Benefits

- ✓ Write Python Code instead of SQL
- ✓ Abstract away Vendor Syntax

Downsides

- ✓ Lot's of “magic” hides the truth
- ✓ Cannot leverage native vendor features

What is an ORM?

OO Constructs

- ✓ Classes == Tables
- ✓ Instance == Row
- ✓ Instance Attributes == Columns

Class (Person)

Instances (2)

Attributes (4)

first	last	height	age
Jimmy	Johnson	74"	33
Willis	Jones	72"	40

What is an ORM?

Simple Python Class Example

```
class Person(ORMSuperClass):
```

```
    def __init__(self, first, last, height, age):
```

```
        self.first = first
```

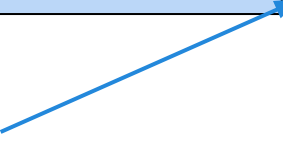
```
        self.last = last
```

```
        self.height = height
```

```
        self.age = age
```

first	last	height	age
Jimmy	Johnson	74"	33
Willis	Jones	72"	40
Cameron	Goodale	73"	36

Person(Cameron, Goodale, 73, 36)



Design Patterns

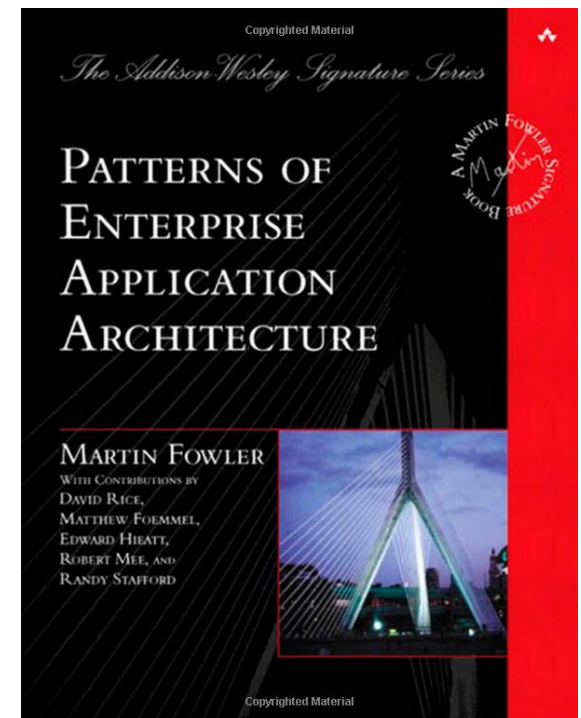
Patterns to describe common problems

- ✓ Data mapper (SQLAlchemy)
- ✓ Active Record (Ruby on Rails)

sources:

http://en.wikipedia.org/wiki/Data_mapper_pattern

http://en.wikipedia.org/wiki/Active_record_pattern



Final Warning

Learn SQL

ORMs hide A LOT

You still need good Database Design

- ✓ Indexes
- ✓ Views
- ✓ Normalized vs. De-normalized
- ✓ Spatial Queries

Thank you

slides: <http://bit.ly/iepSQLA>